НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«На правах рукопису»
УДК 004.042

«До захисту допущено»
Науковий керівник кафедри
_____ І.А. Дичка
«___»_____2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 Інженерія програмного забезпечення

на тему: «Спосіб та програмне забезпечення для аналізу та прогнозування коливань курсу криптовалют»

Виконав:
студент VI курсу, групи КП-71мп
Мухін Юрій Юрійович _____

Керівник:
Старший викладач кафедри ПЗКС, к.т.н.,
Хіцко Я.В. _____

Рецензент:
Професор кафедри обчислювальної техніки, д.т.н.,
Жабін В. І. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2018 року

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Криптовалюта – вид цифрової валюти, облік якої заснований на асиметричному шифруванні і застосуванні різних криптографічних методів захисту.

Тренд – напрямок переважного руху показників. Зазвичай розглядається в рамках технічного аналізу, де мають на увазі спрямованість руху цін або значень індексів.

TimerJob – це програмний комплекс, який налаштовується та виконується ітеративно через певний проміжок часу, виконуючи один і той самий алгоритм тільки з актуальними параметрами.

Кінцевий користувач – це користувач, який використовує систему.

ШПФ – Швидке перетворення Фур’є.

ЗШПФ – Зворотне швидке перетворення Фур’є.

ДПФ – дискретне перетворення Фур’є.

ПЗ – програмне забезпечення.

ВСТУП

Сьогодні спостерігається експонентне зростання інтересу до криптовалюти, що зумовлено розвитком цієї сфери в розробці програмного забезпечення, таких як інтернет-біржі, різноманітні системи для сплати, які дозволяють проводити платежі на різноманітних криптовалютах, зі змогою автоматичного конвертування валют. Внаслідок таких можливостей сьогодні досить гострою є проблема швидких коливань курсу, на яких інвестори чи трейдери втрачають свої кошти.

В той же час криптовалюти отримують все більше інвестицій та мають багато плюсів, також криптовалюти являють собою децентралізоване середовище, що в свою чергу гарантує безпеку.

Криптовалюти мають значні коливання курсу в різні проміжки часу, що і є головною проблемою, оскільки ціна на одиницю формується з багатьох факторів, які мають значні зміни за короткий проміжок часу.

З плюсів можна виділити комісії, які практично не залежать від переданої суми або місця відправлення та прийому. Часто транзакції безкоштовні. Комісія потрібна лише при створенні технічно великих транзакцій, які завантажують мережу або ж при переведенні незначних сум, щоб уникнути атаки мережі спамом з великою кількістю безглуздих транзакцій, що теж є зручним з точки зору безпеки.

Автоматизована система для аналізу криптовалюти являє собою програмний комплекс, який дозволяє збирати та аналізувати дані в реальному часі, та зберігати результати. Також є боти-трейдери (програмний додаток який зазвичай розробляється для вузького кола людей), що має можливість закуповувати та продавати валюту автоматично, зважаючи на зміни курсу, але такі операції з біржею можуть призвести до блокування акаунту.

В цілому все, що доступно звичайному користувачу, це звичайні підходи з аналізу валют в розрізі економіки (в основній частині

використовуються графічні подання статистики). Тому було вирішено розробити автоматичну систему для аналізу та зіставлення прогнозу, яка б сповіщувала користувачів про зміни тренду, що значно покращить та виведе на новий рівень безпеки сам процес взаємодії з криптовалютою.

1. ПРОБЛЕМИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ АНАЛІТИКИ ТА ЗІСТАВЛЕННЯ ПРОГНОЗУ КОЛИВАННЯ КУРСУ

1.1. Аналіз підходів до аналітики коливання курсу криптовалют

Існують два основних способи аналізу фінансових і товарних ринків: фундаментальний і технічний. Під фундаментальним аналізом розуміється використання макроекономічних індикаторів стану економіки (показники ВВП і ВНП, безробіття і зайнятості, інфляції та грошової маси, національного доходу). За ступенем важливості, основні показники, які використовуються в якості фундаментальних чинників, можна розбити на три групи.

До першої групи належать:

- динаміка валового національного (внутрішнього) продукту: при його зростанні курс національної валюти зростає;
- дефіцит торгового балансу: при його зростанні, як правило, курс національної валюти знижується;
- індекси інфляції - індекс споживчих цін і індекс оптових цін: при зростанні інфляції курс національної валюти, як правило, знижується:
- офіційні облікові ставки: при їх зростанні курс національної валюти, як правило, зростає;
- дані з безробіття або зайнятості.

Зростання безробіття (падіння зайнятості), як правило, супроводжується падінням курсу національної валюти. Однак для кожної з країн на сьогодні існують офіційні дані щодо ефективного відсотка безробіття, тобто допустимий і навіть бажаний рівень для розвитку економіки. На сьогоднішній день ці розміри коливаються від 3 до 7% всього працездатного населення в залежності від країни[1].

До другої групи належать фактори:

- розміри роздрібних продаж: при їх зростанні курс національної валюти може зростати;
- розміри будівництва житлової нерухомості: при його зростанні курс може зростати;
- індекс виробничих цін: їх зростання може викликати спадання
- курсу;
- індекс промислового виробництва: його зростання може
- сприяти зростання курсу;
- продуктивність в економіці: падіння продуктивності може викликати падіння курсу.

До третьої групи належать дані, які можна отримувати в режимі реального часу:

- індекси акцій (з різноманітних бірж): зростання цих індексів говорить про хороший стан національної економіки і підвищує попит на національну валюту даної країни;
- динаміка цін державних облігацій – збільшення попиту на державні цінні папери, і як наслідок наступне за цим підвищення їх цін, як правило, супроводжується зростанням курсу національної валюти.

Про показники першої і другої груп можна отримувати дані з інформаційних агентств (Reuters, Dow Jones, AFZ, Knight Ridder). Як джерела також можливе використання статистичних публікацій міжнародного статистичного інституту, перш за все Bulletin of the International Statistics Institute, Annual Review of Statistics та інші.

Аналіз фундаментальних факторів можна звести до розрахунку синтетичного показника, розробленого автором рівня економічного положення ринку[2].

Якщо темпи зростання економічного рівня однієї країни за певний період перевищують цей показник за той же період в іншій країні, то це супроводжується, перш за все, відповідним посиленням національної валюти першої країни, що звісно відбивається на динаміці цін.

Фундаментальний аналіз вивчає рух цін на макроекономічному рівні. Він може сприяти визначенню головного ринкового тренду, однак для визначення конкретного моменту здійснення угоди фундаментального аналізу часто буває недостатньо. У цій області застосовується технічний аналіз.

Технічним часто називають графічний метод аналізу ринку, тому що в даному виді аналізу береться в розрахунок тільки графік руху ціни і обсягу торгів з минулого до теперішнього моменту без урахування всіх інших факторів. Основним принципом технічного аналізу є твердження, що будь-який фактор, що впливає на ціну: економічний, політичний або психологічний, вже врахований ринком і включений в ціну. Важливим положенням є і те, що ринок розвивається направлено: ціни або ростуть, або падають, або знаходяться в нейтральному діапазоні. Тому виявлення тренду або переважаючого руху цін і складає основну базу технічного аналізу і запорука успішної торгівлі базується саме на цьому принципі[3].

Існують три види трендів: зростаючий, спадаючий, горизонтальний. Зростаючий, або «бичачий», тренд характеризується тим, що нижні коливання ціни ринку підвищуються (рис.1). Лінія, що обмежує такий тренд знизу і проходить через мінімальні значення, називається лінією тренду.

При зростаючому тренді важливо мати кордон саме знизу, оскільки в цьому випадку робляться ставки на підвищення ціни. Перетин лінії тренду, яка обмежує криву ціни знизу, може сигналізувати про те, що загальний тренд на підйом ціни або слабшає, або взагалі змінює напрямок.

Фундаментальний аналіз - це термін для визначення методів прогнозування ринкової вартості компанії на основі аналізу його фінансових та виробничих індикаторів. Фундаментальний аналіз

використовується інвесторами для оцінки вартості компанії (або її акцій), яка відображає стан справ у компанії, рентабельність її діяльності. Цей аналіз аналізує фінансові показники компанії: дохід (прибуток перед інтересами, податки, амортизація та амортизація), чистий прибуток, чиста вартість, зобов'язання, рух грошових коштів, сума сплачених дивідендів та виробничі показники компанії. "Внутрішня цінність" в більшості випадків не співпадає з ціною акцій товариства, яка визначається співвідношенням пропозиції та попиту на фондовому ринку. Інвестори, які використовують фундаментальний аналіз у своїй діяльності, в першу чергу цікавляться ситуаціями, коли "внутрішня вартість" акцій компанії перевищує ціну акцій на біржі. Такі запаси вважаються недооціненими та є потенційними об'єктами інвестування. Купуючи недооцінені акції, інвестори стверджують, що в умовах неефективності ринку ціна акцій на фондовому ринку буде мати тенденцію до "внутрішньої вартості", тобто у випадку недооцінених акцій вона буде зростати. Це твердження є зворотним до постулату технічного аналізу, який стверджує, що вся суттєва інформація негайно і повністю відображає ринкову вартість цінних паперів. І цей принцип заперечує ідею фундаментального аналізу. В свою чергу використовувати цей тип аналізу для автоматизації процесу неможливо, тому що він базується на простому зборі інформації та усвідомленні майбутніх або минулих подій.

Технічний аналіз - набір інструментів для прогнозування можливих змін цін на основі моделей змін цін у минулому за аналогічних обставин. Основною базою є аналіз цінових схем - графіків та фондової біржі. Теоретично, технічний аналіз можна застосовувати на будь-якому ринку[4].

Але найчастіше технічний аналіз використовується на високоліквідних вільних ринках, таких як біржі. У технічному аналізі існує безліч інструментів і методів, але всі вони ґрунтуються на одному припущенні: динаміка руху цін визначається психологією поведінки учасників ринку, які в подібних обставинах під впливом людських

інстинктів - жадібності, стадного інстинкту, страху, тощо - ведуть себе так само, як потоки попиту та пропозицій. Це дозволяє зіставляти прогноз руху цін.

Технічний аналіз не враховує причин, через які ціна змінює свій напрямок (наприклад, через низьку прибутковість акцій, коливання цін на інші товари або зміни в інших умовах), але враховує лише те, що ціна рухається в одному напрямку або в іншому або певним чином, наприклад, якийсь час перебуває в межах певного цінового діапазону.

Різні методи можуть також враховувати ставки, кількість відкритих позицій, обсяг поданих заяв на купівлю / продаж та інші.

Другий тип аналізу, який підходить, це аналіз на основі даних графів та коефіцієнтів фондової біржі.

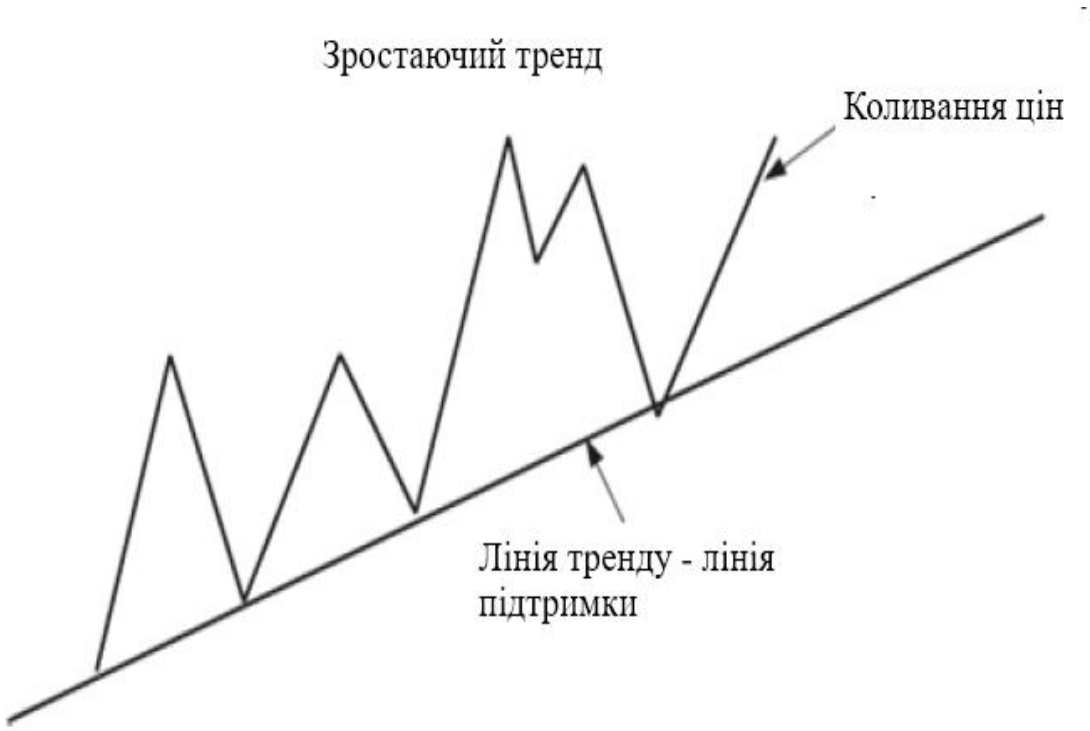

Рис. 1. Зростаючий тренд

Лінія тренду, що обмежує криву ціни знизу, називається лінією підтримки. Спадаючий, або «ведмежий», тренд характеризується тим, що максимальні ціни коливань ринку знижуються (рис. 2)[5].

Рис. 2. Спадаючий тренд

При спадному тренді слід приділяти увагу до лінії тренду, яка обмежує криву динаміки цін зверху і називається лінією опору. При існуванні «ведмежого» ринку встановлюються ціни на зниження, тому для трейдерів важливе тільки обмеження цін зверху, тому що чим нижче опуститься ціна, тим краще для трейдера.

Перетин, або пробиття лінії опору, попереджає трейдера про можливе ослаблення тренду, або навіть його зміні.

Особливий інтерес для трейдерів представляють так звані канали, коли для чітко вираженого тренду існують, як хороші лінії підтримки, так і опору (рис. 3). При цьому тренд дає можливість прогнозувати як нижні, так і верхні їх рівні.

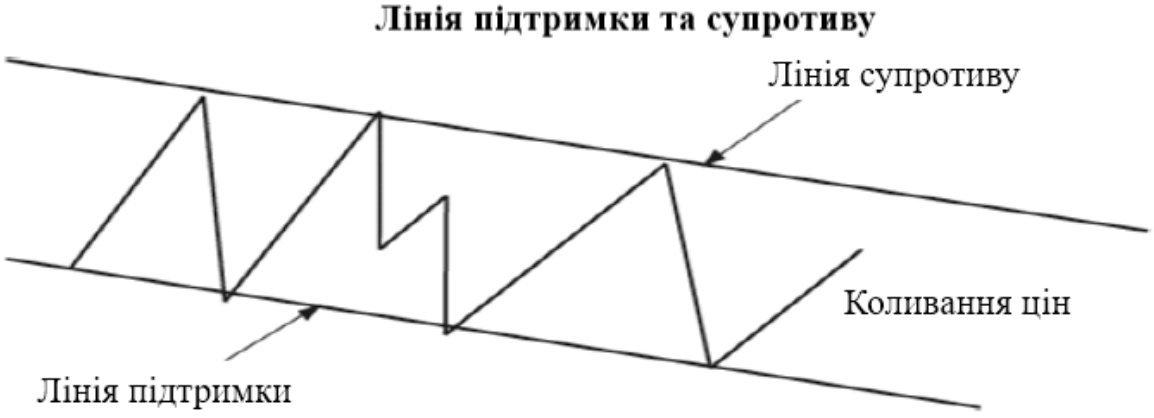

Рис. 3. Лінія підтримки та супротиву

Третій тип тренду - горизонтальний, він показує, що ціни коливаються в горизонтальному діапазоні[6]. Для цього типу тренду теж існують лінії підтримки і опору, але відсутній явно виражений рух цін вгору або вниз (рис. 4).

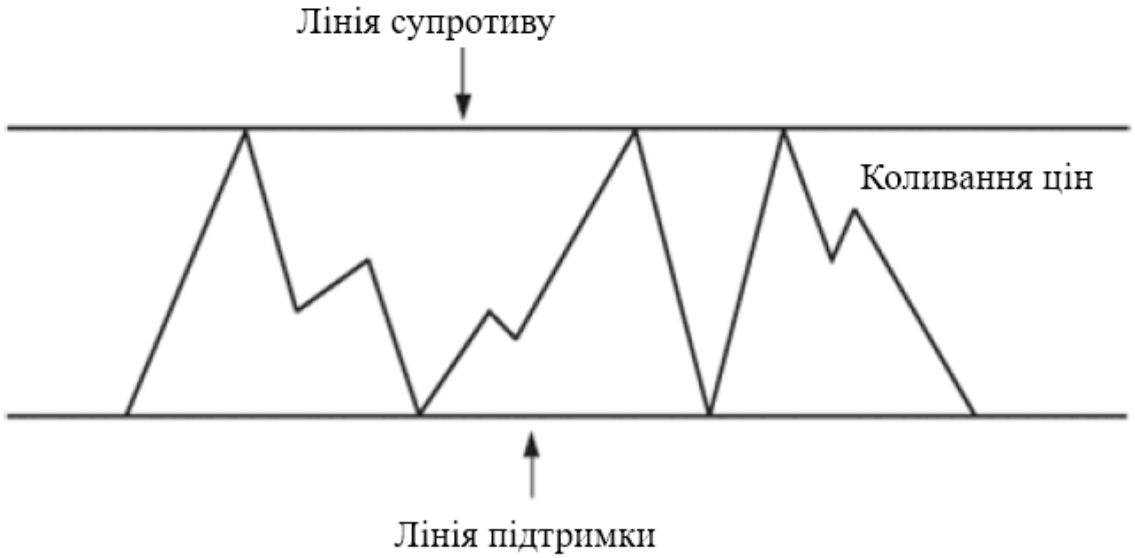

Рис. 4. Горизонтальний тренд

Метод нанесення на графік зміни ціни за певний проміжок часу називається чартом (англ. chart – діаграма). Відображення чартів може бути здійснено різноманітними способами, починаючи від форми подання даних до їх частоти, періоду та інше (рис. 5).

Рис. 5. Відображення чарту на криптовалютній біржі Bittrex

Чарти бувають різних типів, в залежності від методу відображення, можна отримати швидкий доступ до згрупованих даних в залежності від методу. Лінійний чарт (рис. 5) представляє собою звичний графік коливання цін в залежності від часу.

CandleStick (японські свічки) - штрихове відображення ціни, якщо ціна відкриття(на початок дня) нижче ціни закриття(на кінець дня), тіло свічі буде зафарбовано червоним кольором, якщо ціна відкриття буде вище ніж ціна закриття, то свічка буде зафарбована зеленим кольором (рис. 6).

Лінія, яку ціна не в змозі пробити вгору, називається лінією супротиву, а лінія, яку ціна не зможе пробити донизу, називається лінією підтримки. Ці лінії малюються на кінцях стовпця різниці динаміки цін. Вони дають можливість здійснювати покупки-продажу, попереджають про події які відбулися за день, та дозволяють аналізувати ситуацію на ринку.

Рис. 6. Відображення CandleStick на криптовалютній біржі Bittrex

Канал, який утворюється в результаті проведення паралельних ліній (супротиву, опору), являє собою оптимальний діапазон торгових змін. Напрямок каналу вниз або вгору визначає тенденцію ринку (тобто тренд) (рис. 7). При висхідній тенденції тренд зростає, при низхідній - знижується. Коли немає тенденції, канал має горизонтальне положення[7].

Рис. 7. Визначення тренду по напрямку каналу на криптовалютній біржі Bittrex

1.2. Аналіз методів зіставлення прогнозу валютних коливань

Зворотні фігури (reversal patterns). Зворотна фігура свідчить про глобальну зміну тренду. Використовувати зворотні фігури є сенс тільки при наявності чітко вираженого попереднього тренду.

Подвійна вершина (дно), зворотна фігура для зростаючого тренду має вигляд букви М. Подвійна вершина є сигналом більш слабким, ніж потрійна вершина (рис. 8), для знижувального тренду, має вигляд букви W.

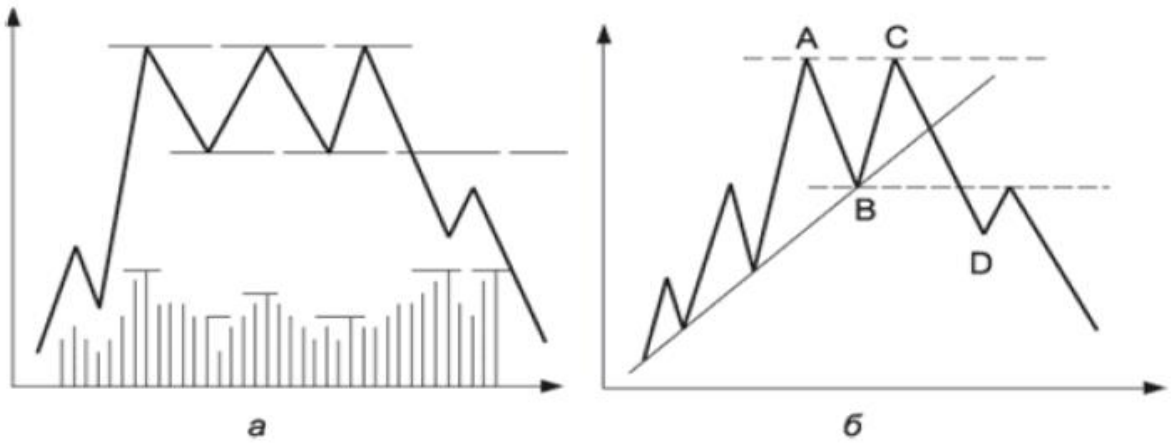

Рис. 8. Спадаючий тренд:
а – при трійному піку, б – при двійному піку

Голова і плечі (head & shoulders). Зворотна фігура для зростаючого тренду. Голова і плечі – графічна фігура, коли динаміка ціни досягає піку і падає, після піднімається вище попередньої максимальної точки і знову

падає, знову піднімається, але не досягає другий максимальної точки і знову падає.

При цьому:

- плечі - перша і третя максимальні точки;
- голова - другий пік графіка;
- лінія шиї - лінія, що з'єднує плечі.

Пробій ціною лінії шиї є сигналом про початок сильного руху ринку проти попередньої тенденції. Зазвичай ціни проходять відстань від точки пробою не менше відстані від вершини голови до лінії шиї, врахованого по вертикалі.

Перевернуті голова і плечі (inverse head & shoulders). Зворотна фігура для спадаючого тренду; фігура, отримана дзеркальним відображенням фігури «голова і плечі».

Збита вершина (blow-off top). Зворотна фігура показує стрімке зростання і спадання цінового графіка.

Потрійна вершина (triple top). Зворотна фігура для зростаючого тренду. Потрійна вершина є сигналом більш слабким, ніж фігура «голова і плечі »

Рис. 9. Голова та плечі:

1 – вершина лівого плеча; 2 – вершина голови; 3 – вершина правого плеча; 4 – лінія шиї.

Потрійне дно (triple bottom). Зворотна фігура для спадаючого тренду. Потрійне дно як сигнал, є більш слабким, ніж перевернуті голова і плечі.

Фігури продовження тренду. Фігури, які свідчать про продовження попереднього тренду, відносяться до розряду корегуючих, фігури продовження тренду мають сенс тільки при наявності чітко вираженого попереднього тренду. Відрізняються такі фігури продовження тренду наступними категоріями[8]:

- зростаючі вершини (ascending tops) - індикатор підвищення - фігура, у якій кожна наступна верхня відмітка ціни вище попередньої;
- зростаючі денця (ascending bottoms) - індикатор підвищення - фігура, у якій кожна наступна нижня відмітка ціни вище попередньої;
- спадаючі вершини - індикатор зниження, діаграма зміни цін, у якій кожна наступна верхня відмітка нижче попередньої;
- удар (thrust) - ситуація, коли після завершення трикутника ціна різко змінюється. Мінімальний розмір поштовху дорівнює довжині найбільшої хвилі трикутника.
- прапори та вимпели - є частими моделями фігур продовження, які грають роль корекцій досить динамічних трендів (рис. 10). Вони мають багато спільного в динаміці. Обидві моделі виникають після енергійного руху ринку, зображеного на графіку у вигляді майже прямої лінії. Сама постать прапора / вимпела є паузою в розвитку тренду, в проміжку якої ціна тримається майже на одному рівні. Вимпел, по суті, представляє собою маленький трикутник (на «держаку»), а прапор – нетривалий ціновий канал (також на держаку у вигляді попереднього руху діаграми). Після прориву фігури тренд продовжує попередню тенденцію, проходячи відстань, приблизно рівній тій, що мала місце до прориву.

Рис. 10. Фігура вимпел

Прапор та вимпел є надійними фігурами продовження тренду, коли вони спрямовані проти тренду (так що прорив відбувається проти нахилу прапора чи вимпела).

Отже, прапор, нахилений вниз, сигналізує про «бичачий» тренд, а спрямований вгору - «ведмежий». Активність торгівлі до фігури велика, потім падає, і знову різко зростає після прориву. Видима закономірність полягає також у тому, що на падаючому ринку прапори і вимпели формуються швидше, ніж на зростаючому. Появі цих фігур на енергійно зростаючому ринку часто слугують розриви тренду. Слід зазначити, що прапори і вимпели, спрямовані не проти, а за попереднім трендом, є фігурами звороту.

Фігури продовження тренду мають менший період формування, ніж зворотні фігури, тому їх можна аналізувати не тільки на тижневих і місячних проміжках часу, але і на денних. Так само, як і для зворотних фігур, для фігур продовження тренду важливі обсяги, які падають при рухах проти тренду і зростають в направленні тренду.

На відміну від зворотних фігур, фігури продовження значно рідше опиняються недостовірними. Як показує практика, найбільш часто використовуваною фігурою технічного аналізу є трикутник. Трикутник як

фігуру технічного аналізу можна розглядати з двох різних сторін, але не суперечливих одна одній:

- що відбудеться всередині даної фігури;
- що станеться після завершення даної фігури.

Трикутник - фігура продовження тренду (рис.11), пересічні лінії опору і підтримки якої чітко виражені. Залежно від обмежуючих ліній розрізняють види трикутника: східний, висхідний, низхідний, розхідний.

В залежності від положення вхідного сигналу (тобто позиції тренду на даний момент часу), та відношення до однієї з обмежуючих ліній, можна робити висновок про зміну чи незмінність тренду на певний проміжок часу. Що в свою чергу дає певну інформацію, з якою можна працювати та робити певні висновки щодо зміни напрямку тренду. Далі розглянуто різні типи фігур (в даному випадку трикутник), які мають різні форми, та сигналізують про різні тенденції, щодо зміни положень на ринку.

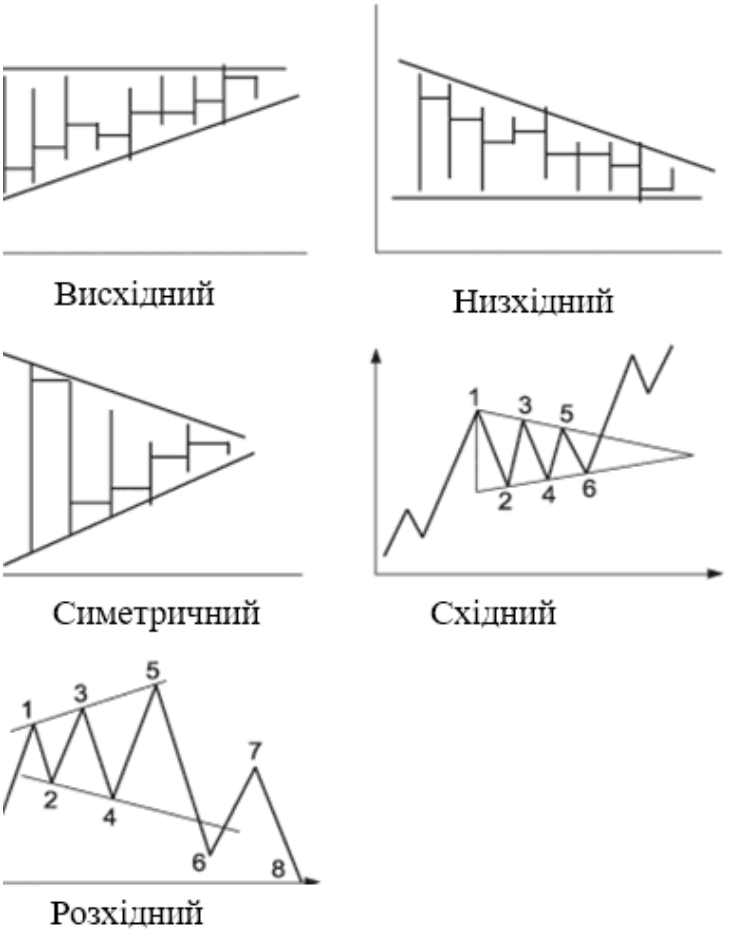

Рис. 11. Фігура трикутник

Прорив трикутника відбувається лише у випадку, якщо ціна закриття торгів була зафіксована за межами лінії консолідації (тобто вище, або ж нижче однієї із сторін трикутника) проте, якщо бік стовпчика графіка був перетнутий, в свою чергу, це випадок наддасть недостовірний сигнал. Після прориву трикутника графік зазвичай повертається до пробитої сторони (лінії консолідації). «Бичачі» трикутники часто дають в результаті недостовірний сигнал, особливо якщо графік пройшов занадто далеко до вершини трикутника. При цьому, як уже зазначалося, вихідною точкою для аналізу є масштаб ціни та часу, в якому працює трейдер.

Трикутник використовується при невпевненості трейдерів щодо напрямку руху ринку або при очікуванні якої-небудь події, яка викликає знову ж таки невпевненість трейдерів.

Змінна середня (moving average) – найстаріші, найпростіші і тому найчастіше використовувані показники. Даний показник є середнім арифметичним цін закриття певного числа попередніх періодів часу, включаючи також поточний період.

Дотик або ж перетин значеннями цін ліній середніх буде сигналізувати про можливий розворот тренду, особливо якщо діаграма закриття цін, також пересікає лінії середніх. Знаходження значень цін між лініями середніх говорить про невизначеність руху. Перетин лінією цін швидкої лінії може передбачати короткочасну зміну напрямку тренду, в той час як перетин лінією цін повільних ліній говорить про сильний і довгостроковий розворот.

Межі Боллінгера (bollinger bound). При побудові даного індикатора розраховуються не тільки середнє відхилення, але і стандартне відхилення цієї ж послідовності цін закриття ринку.

1.3. Сучасні системи для аналітики та зіставлення прогнозу валютних коливань

Автоматизовані системи для аналізу криптовалюти являють собою програмні комплекси, які дозволяють збирати та аналізувати дані в реальному часі, та зіставляти на їх основі певні прогнози щодо зміни тенденцій ринку. Але автоматичних систем для збору та зіставлення прогнозу для криптовалюти на даний момент не існує.

В цілому все, що доступно користувачу, це звичайні підходи з аналізу валют в розрізі економіки (в основній частині використовуються графічні подання статистики). Тому було вирішено розробити автоматичну систему для аналізу та прогнозування, сповіщення користувачів про зміни тренду, що значно покращить та виведе на новий рівень безпеки сам процес взаємодії з даним типом валют.

1.4. Висновки до розділу 1

В даному розділі був проведений аналіз існуючих підходів аналізу та зіставлення прогнозів валютних ринків. Було проаналізовано їх переваги та недоліки.

Було розглянуто методи аналізу та зіставлення прогнозів та виділено основні різновиди аналізу: фундаментальний та технічний, але надалі буде використано лише підходи до технічного аналізу, так як саме цю частину можна інтерпретувати та автоматизувати за допомогою комп’ютерних алгоритмів. Такі алгоритми дозволять розробити спосіб до аналізу та зіставлення прогнозу на певний проміжок часу.

2. СПОСІБ АНАЛІЗУ КУРСУ КРИПТОВАЛЮТИ ТА РОЗРОБКА МЕТОДУ ДЛЯ ЗІСТАВЛЕННЯ ПРОГНОЗУ НА ПЕВНОМУ ПРОМІЖКУ ЧАСУ

2.1 Аналіз курсу криптовалюти

Дані про курс криптовалюти можна отримувати з криптовалютних бірж. Їх на даний момент досить багато, тому потрібно було обрати один конкретний варіант, звідки б можна було отримувати найактуальніші дані.

В той же час, щоб спосіб надання цих даних був зручним, та вкладався в архітектуру та обсяг даних розроблюваного програмного забезпечення. Для обрання біржі, використовувалися такі критерії:

- якими валютами потрібно оперувати - чим більш екзотичний вибір монет, тим менше платформ підходять. В свою чергу, слабкою може виявитися і популярність ринку, так що продати / купити велику партію буде складно;
- наскільки надійна платформа – чим надійніша платформа, тим більш надійними будуть і дані, які повинні надходити безперервно;
- спосіб отримання даних – потрібен зручний сервіс отримання даних для сторонніх систем.

В результаті аналізу існуючих бірж, на основі описаних вище критеріїв було обрано біржу Bittrex (рис. 11). Основними її перевагами є:

- представлено понад 180 альткойнів, зведених в більш ніж 400 торгових пар;
- хороший рівень безпеки облікового запису з двоетапною авторизацією;
- відсутність форс-мажорних ситуацій та інформації про зломи або розкрадання;
- зручний спосіб отримання даних, через API в форматі JSON, такий спосіб дозволяє інтегруватися з будь яким додатком, через спосіб серіалізації та десеріалізації даних.

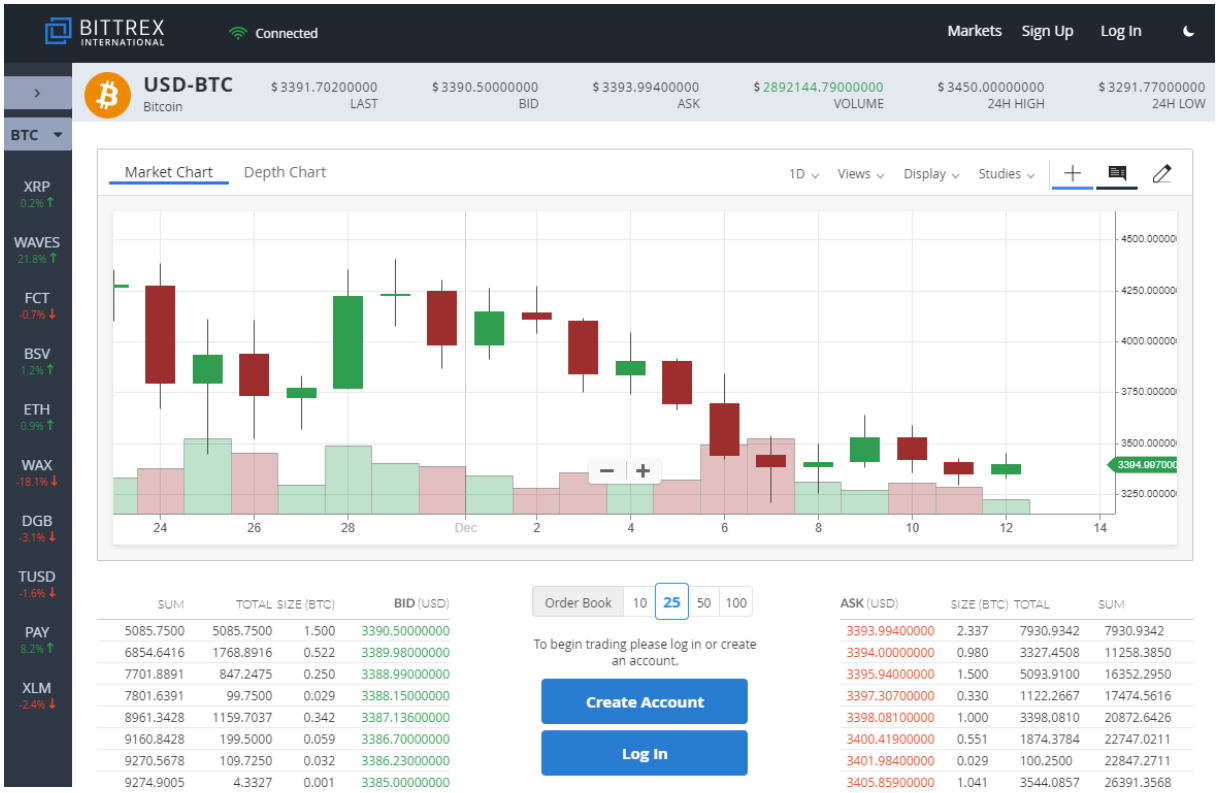

Рис. 12. Сторінка біржі Bittrex

Для збору та збереження актуальних даних про курс криптовалюти на біржі, було використано публічне API біржі. Bittrex надає простий і потужний API REST, який дозволяє вам програмно виконувати практично всі дії, які ви можете виконувати з веб-інтерфейсу.

Всі запити використовують тип вмісту *application/json*, та відправляються через протокол *https*. Базова URL-адреса - *https://bittrex.com/api/{version}/*. Всі запити - це запити GET, і всі відповіді надходять як типовий об'єкт відповіді, результаті буде знаходитися у полі результату. Також є прапорець успіху, щоб переконатися, що виклик API вийшов успішно, та дані були отримані вірно.

На даний момент Bittrex обмежує кількість запитів на 500 відкритих і 200 000 замовлень на день. Також при необхідності, Bittrex може змінювати ці налаштування при налагодженні системи.

Bittrex API розбите на три різні групи:

- Публічне - загальнодоступна інформація доступна без ключа API;

- Ринкове - для програмної торгівлі криптовалютою;
- Персональне - для керування вашим обліковим записом.

Публічне API:

1. Запит: *https://bittrex.com/api/v1.1/public/getmarkets*

Використовується для отримання відкритих та доступних торгових ринків у Bittrex разом з іншими метаданими.

Відповідь:


```
{
    "success" : true,
    "message" : "",
    "result" : [{
            "MarketCurrency" : "LTC",
            "BaseCurrency" : "BTC",
            "MarketCurrencyLong" : "Litecoin",
            "BaseCurrencyLong" : "Bitcoin",
            "MinTradeSize" : 0.01000000,
            "MarketName" : "BTC-LTC",
            "IsActive" : true,
            "Created" : "2014-02-13T00:00:00"
        }, {
            "MarketCurrency" : "DOGE",
            "BaseCurrency" : "BTC",
            "MarketCurrencyLong" : "Dogecoin",
            "BaseCurrencyLong" : "Bitcoin",
            "MinTradeSize" : 100.00000000,
            "MarketName" : "BTC-DOGE",
            "IsActive" : true,
            "Created" : "2014-02-13T00:00:00"
        }
```

```
   ]
}
```


2. Запит: *https://bittrex.com/api/v1.1/public/getcurrencies*

Використовується для отримання всіх підтримуваних валют у Bittrex разом з іншими метаданими.

Відповідь:


```
{
        "success" : true,
        "message" : "",
        "result" : [{
                        "Currency" : "BTC",
                        "CurrencyLong" : "Bitcoin",
                        "MinConfirmation" : 2,
                        "TxFee" : 0.00020000,
                        "IsActive" : true,
                        "CoinType" : "BITCOIN",
                        "BaseAddress" : null
                }, {
                        "Currency" : "LTC",
                        "CurrencyLong" : "Litecoin",
                        "MinConfirmation" : 5,
                        "TxFee" : 0.00200000,
                        "IsActive" : true,
                        "CoinType" : "BITCOIN",
                        "BaseAddress" : null
                }
   ]
}
```

3. **Запит**: *https://bittrex.com/api/v1.1/public/getticker*

Використовується для отримання поточних значень тиків для ринку.

Відповідь:


```
{
    "success" : true,
    "message" : "",
    "result" : {
        "Bid" : 2.05670368,
        "Ask" : 3.35579531,
        "Last" : 3.35579531
    }
}
```


4. **Запит**: *https://bittrex.com/api/v1.1/public/getmarketsummaries*

Використовується для отримання останнього цілодобового підсумку всіх активних ринків.

Відповідь:


```
{
    "success" : true,
    "message" : "",
    "result" : [{
            "MarketName" : "BTC-888",
            "High" : 0.00000919,
            "Low" : 0.00000820,
            "Volume" : 74339.61396015,
            "Last" : 0.00000820,
            "BaseVolume" : 0.64966963,
            "TimeStamp" : "2014-07-09T07:19:30.15",
            "Bid" : 0.00000820,
            "Ask" : 0.00000831,
            "OpenBuyOrders" : 15,
```

"OpenSellOrders" : 15,
"PrevDay" : 0.00000821,
"Created" : "2014-03-20T06:00:00",
"DisplayMarketName" : null
}, {
"MarketName" : "BTC-A3C",
"High" : 0.00000072,
"Low" : 0.00000001,
"Volume" : 166340678.42280999,
"Last" : 0.00000005,
"BaseVolume" : 17.59720424,
"TimeStamp" : "2014-07-09T07:21:40.51",
"Bid" : 0.00000004,
"Ask" : 0.00000005,
"OpenBuyOrders" : 18,
"OpenSellOrders" : 18,
"PrevDay" : 0.00000002,
"Created" : "2014-05-30T07:57:49.637",
"DisplayMarketName" : null
}
]
}

5. Запит:

https://bittrex.com/api/v1.1/public/getmarketsummary?market=btc-ltc

Використовується для отримання останнього 24-годинного резюме конкретного ринку.

Відповідь:

{
"success" : true,
"message" : "",

```
        "result" : [{
                        "MarketName" : "BTC-LTC",
                        "High" : 0.01350000,
                        "Low" : 0.01200000,
                        "Volume" : 3833.97619253,
                        "Last" : 0.01349998,
                        "BaseVolume" : 47.03987026,
                        "TimeStamp" : "2014-07-09T07:22:16.72",
                        "Bid" : 0.01271001,
                        "Ask" : 0.01291100,
                        "OpenBuyOrders" : 45,
                        "OpenSellOrders" : 45,
                        "PrevDay" : 0.01229501,
                        "Created" : "2014-02-13T00:00:00",
                        "DisplayMarketName" : null
                }
    ]
}
```


6. **Запит**: *https://bittrex.com/api/v1.1/public/getticker*

Використовується для отримання замовлення для певного ринку.

Відповідь:


```
{
        "success" : true,
        "message" : "",
        "result" : {
                "buy" : [{
                                "Quantity" : 12.37000000,
                                "Rate" : 0.02525000
                        }
                ],
```

```
            "sell" : [{
                        "Quantity" : 32.55412402,
                        "Rate" : 0.02540000
                    }, {
                        "Quantity" : 60.00000000,
                        "Rate" : 0.02550000
                    }, {
                        "Quantity" : 60.00000000,
                        "Rate" : 0.02575000
                    }, {
                        "Quantity" : 84.00000000,
                        "Rate" : 0.02600000
                    }
            ]
        }
}
```


7. **Запит**

https://bittrex.com/api/v1.1/public/getmarkethistory?market=BTC-DOGE

Використовується для отримання останніх операцій, що відбулися для певного ринку.

Відповідь:


```
{
    "success" : true,
    "message" : "",
    "result" : [{
                "Id" : 319435,
                "TimeStamp" : "2014-07-09T03:21:20.08",
                "Quantity" : 0.30802438,
```

```
            "Price" : 0.01263400,
            "Total" : 0.00389158,
            "FillType" : "FILL",
            "OrderType" : "BUY"
        }, {
            "Id" : 319433,
            "TimeStamp" : "2014-07-09T03:21:20.08",
            "Quantity" : 0.31820814,
            "Price" : 0.01262800,
            "Total" : 0.00401833,
            "FillType" : "PARTIAL_FILL",
            "OrderType" : "BUY"
        }, {
            "Id" : 319379,
            "TimeStamp" : "2014-07-09T02:58:48.127",
            "Quantity" : 49.64643541,
            "Price" : 0.01263200,
            "Total" : 0.62713377,
            "FillType" : "FILL",
            "OrderType" : "SELL"
        }, {
            "Id" : 319378,
            "TimeStamp" : "2014-07-09T02:58:46.27",
            "Quantity" : 0.35356459,
            "Price" : 0.01263200,
            "Total" : 0.00446622,
            "FillType" : "PARTIAL_FILL",
            "OrderType" : "BUY"
        }
]
```

}

Існує всього два підходи до аналізу валютних коливань, які базуються на певних категоріях аналізу , а саме:

• Фундаментальний аналіз;

• Технічний аналіз.

У попередньому абзаці було виявлено, що для того, щоб мати змогу автоматизувати процес зіставлення прогнозу, потрібно покладатися лише на аспекти технічного аналізу та працювати лише з графіками та даними. Також варто розуміти, що таке діаграма і які дані можна з неї отримати для подальшого дослідження цін. Найчастіше використовуваним показником, а точніше типом діаграми, є японські свічки. Це класове подання цінового руху, яке є одним із найважливіших інструментів для аналізу та зіставлення прогнозу на певний проміжок часу.

Японські свічники є типом інтервальних діаграм, а технічний індикатор використовується в основному для відображення змін у біржових котируваннях, товарних цінах. Таблицю типу «свічки» також називають комбінацією інтервальних та лінійних графіків у тому сенсі, що кожен з його елементів являє собою діапазон змін цін протягом певного часу. Найчастіше він використовується в технічному аналізі ринку. "Свічка" складається з чорного або білого тіла та верхньої / нижньої частини тіла. Верхня і нижня межі тіла «свічки» відображає максимальні та мінімальні ціни за відповідний період. Межі тіла відображають ціну відкриття та закриття.

Якщо ціни загалом зросли, то тіло зелене (колір тіла зелений), нижня межа тіла відображає ціну відкриття, а верхня - ціна закриття. Якщо ціни впали, то тіло буде червоним (колір тіла червоний), верхня межа тіла відображає ціну відкриття, нижня - ціна закриття. Якщо ціни на відкриття / закриття співпадають з найвищим / найнижчим, то відповідно такі тіла не можуть бути відображені. При збіганні цін на відкриття та закриття меж не

сформує потрібну діаграму. Свічка не містить прямих відомостей про рухи цін у відповідному інтервалі часу. Немає жодних ознак того, що максимальний або мінімальний поріг був досягнутий спочатку дня або на прикінці, скільки разів спостерігалося зростання або падіння цін. Наприклад, за наявності двох меж не можна сказати, чи ціна спочатку піднялася чи знизилася. Щоб з'ясувати, потрібно вивчити графік меншого інтервалу часу. Для здійснення подальшого алгоритму найкраще підходить набір із п’яти фігур, голова та плечі (рис. 12), вимпел (рис 13.), прапор (рис. 14), трикутник (рис. 15), потрійний пік (рис. 16).

Рис. 12. Фігура голова та плечі:

Є багато фігур, і крім того, майже всі вони дублюються, та мають альтернативні форми, які називаються зворотними. Подвійний піковий малюнок має зворотний шаблон як подвійний низ те ж саме і стосується трикутника та потрійного піку і так далі.

Саме ці фігури були обрані тому, що показники цих фігур, доволі точно сигналізують про зворотну тенденцію, яка описується за співвідношенням, відповідно, тренд розвертається в протилежному напрямку відносно поточного ходу, що в результаті і є ціллю прогнозування.

Рис. 13. Фігура вимпел

Рис. 14. Фігура прапор

Рис. 15. Фігура трикутник

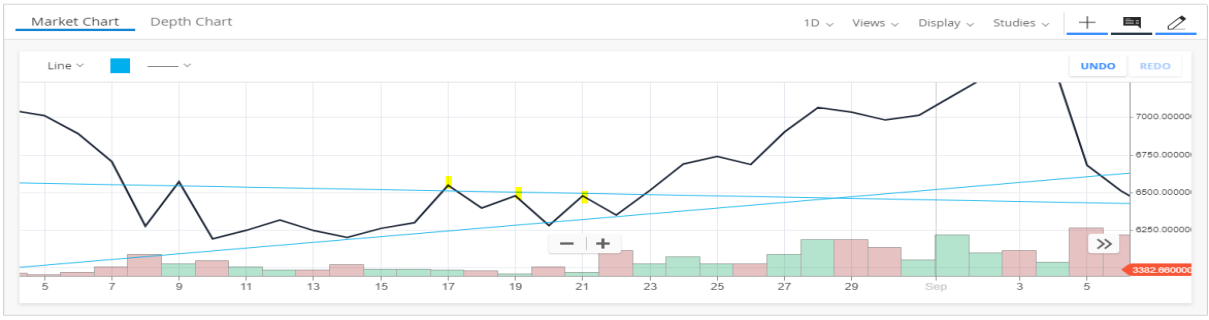

Рис. 16. Потрійний пік

2.2 Розробка методу зіставлення прогнозу базуючись на аналізі графіку курсу за певний проміжок часу

Для розробки базового алгоритму та принципу прогнозування, були використані дані, які були отримані при аналізі існуючих підходів до аналізу валютних коливань.

Далі розкриваються терміни які використовуються при описі алгоритму.

Індекс відносної сили (RSI) - індикатор технічного аналізу, що визначає силу тренду та ймовірність його зміни. Популярність RSI обумовлена простотою його інтерпретації. Індикатор може малювати фігури технічного аналізу - «голова-плечі», «вершина» та інші, які часто аналізують нарівні з графіком ціни.

Ковзаюча середня (MA, середня закачування або середня середня) - це розрахунок точок даних шляхом створення серії середніх різних підмножин набору повних даних для подальшого аналізу. Він також називається рухомим середнім (ММ) або проміжним середнім значенням і являє собою тип фільтра кінцевого імпульсного відгуку. Варіанти включають: прості та сукупні або зважені.

Загалом з вхідних даних отримуємо:

• Рух ціни активу протягом певного періоду часу N у вигляді свічок з кроком K. N та K можуть динамічно змінюватися.

• RSI співвідношення та MA співвідношення.

• Набір фігур, на якому можна проводити аналіз.

Для реалізації потрібен один з типів інтегральних перетворень, а саме згортка. Згортка функцій - операція в функціональному аналізі.

За визначенням, згортка - це математична операція, застосована до двох функцій f і g, що генерує третю функцію, яку іноді можна вважати модифікованою версією однієї з вхідних. По суті, це специфічний вид інтегрального перетворення[9].

Операцію згортки можна інтерпретувати як подібність однієї функції з відображеною та зміщеної копією іншої. Згортку також можна охарактеризувати як вагу однієї функції в тому випадку, якщо інша функція, що відбивається і зміщується, є вагомою.

Таким чином, мінімізуючи функцію графіку ціни за функцію графіку фігури, отримуємо нову функцію, в якій піки стануть точками найбільшої конвергенції реального графіку та моделі. Але також потрібно заздалегідь знати про процес зміни тенденції, щоб отримати прибуток. Для цього було застосовано покроковий алгоритм. Ми розглянемо приклад шаблону голова та плечі.

1. Почнемо з того, що нам потрібно відрізати частину, яка є так званим пост-ефектом і продовженням шаблону, так що після відсічення ми маємо лише ліве плече і голову (рис. 18). Відсікаючи таким способом, що після фіксації в точці відсічення, відповідно до цієї цифри, буде збільшуватися ціна, падіння може тривати до тих пір, поки це необхідно, і це зручний спосіб відстеження переломного моменту для подальшого аналізу, звичайно, для кожної фігури, відсік відбувається по-різному, але завжди в моменти зміни тенденції вгору.

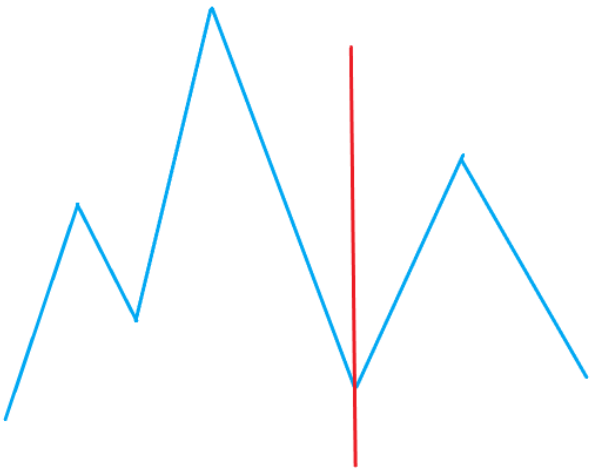

Рис. 17. Відокремлення шаблону

Рис. 18. Результат відокремлення

2. Другий етап - це нормалізація відносних графіків один до одного, всі графіки лежать у площині відносно ціни на *Y* та шкали часу на *X* , але цінова шкала не береться до уваги, тому що для роботи алгоритму потрібен лише напрямок руху цін, а не саме зростання, відповідно. Це дозволяє нормалізувати графік для *Y* без втрати точності результату в майбутньому. Для нормальної взаємодії з графіками цінова поведінкова діаграма для часового інтервалу *N* та діаграма рисунка повинна бути розміщена вздовж осі *Y* піками в інтервалі *N*, а для звичайної моделі максимальна точка приймається як одиниця.

3. Третій етап - комбінація різновидів графічних фігур, що мають нормалізовану функцію (рис. 19), для застосування інтегрального введення потрібно мати графік. Оскільки фігура на реальному графіку може бути розтягнута на невідомий період часу, і в такому випадку визначити її неможливо, тому необхідно створити колекцію функцій (шаблонів), які розтягуються за різними часовими інтервалами, основним фактором створення такої колекції, є часовий інтервал, який буде сигналізувати про зміну ціни, в залежності від оброблюваного ряду.

Почнемо з того, що знайдемо всі моменти, коли функція змінює напрямок "відокремленої фігури". Отримаємо 3 такі точки(рис. 19), а також відправну та звичайно кінцеву точку. Між точками перегину є *n* число

свічок, для генерації групи моделей, просто змінюючи число n від 1 до 5 свічок в інтервалі між кожними точками перфорації функції, це можливо більше, але 5 підходить для коротких торгових позицій.

4. Отримавши нормалізовану функцію руху цін і групу графіків окремої моделі, ми можемо перейти до процесу згортки. Якщо ми проведемо аналогію, шаблони фігур - це фільтри, а діаграма цін є сигналом для фільтрації .

Почнемо з того , що застосуємо інтегральне перетворення згортки, яке було описане вище, для цінових графіків всіх функцій з групи шаблонів і отримаємо нові стартові функції для подальшої роботи. Це покаже нам ступінь охоплення наших «фільтрів» від основного цінового розкладу на весь інтервал початку N.

Якщо t - часовий інтервал свічки, а N - часовий проміжок графіку що описується деякою функцією F, а поточний фільтр описується функцією G_i, то алгоритм виглядає таким чином: кожний інтервал t свічка додається до цінової діаграми, і функція F модифікується, а інтервал N стає $N + t$, потім до цієї функції F застосуємо набір фільтрів, складених за шаблонами, а саме: функцію G_i за процесом згортки і таким чином на виході ми отримуємо значення нової результуючої функції, тому кожного разу t отримаємо нові значення отриманої функції, і чим ближче це значення до одиниці вздовж осі Y, тим більша подібність, тому що значення функції, які ми раніше нормалізували, тим більше шансів, що ми опинимося наприкінці шаблону, і з цією ймовірністю можна сказати, що відбувається з наступним моментом часу t, і таким чином рекурсивний алгоритм продовжується, сигналізуючи про моменти зміни тренду.

Будемо сприймати алгоритм згортки як швидкий алгоритм перетворення Фур’є, який базується на інтегральному перетворенні Фур'є .

Нехай сигнал (наприклад, дані з біржі) і дані N_1 і N_2 будуть відомі, як реальні періоди. Мінімальний період Швидкого перетворення Фур'є без перекриття: $M \geq 3N_1 - 2$. При обчисленні ШПФ , то буде достатньо,

збільшити період до $N = 2^q \geq$ M. Загалом, такі операції необхідно виконати.

1. Заповнити послідовність сигналу N_1 і шаблону N_2 з нульовими зразками до загальної довжини періоду *H*.

2. Виконати прямий суму сигналів ШПФ з довжиною *N* зразків (уявна складова буде дорівнювати нулю).

3. Виконати пряму суму ШПФ шаблону з довжиною *N* зразків (уявна складова дорівнює нулю).

4. Виконати множення самих спектрів сигналу та шаблону. Для цього в цілому необхідно провести *N* комплексних множень.

5. Виконати зворотне ШПФ (ЗПФ) та отримати результати *N* складних множень.

6. Обрати з результатів реальної частини N_1 значення згортки (уявна складова результатів ЗШПФ, очевидно, дорівнює нулю).

В результаті виконання пунктів 1-6, формально використовуючи теорему про еквівалентність згортки в часовій області до множення в частотній області, можна отримати апперіодичну згортку вихідного сигналу N_1з ядром $N_1 = 2N_1 - 1$.

Приклад реалізації ШПФ на мові програмування C#[10]

```
public class FFT
    {
        private static Complex w(int k, int N)
        {
            if (k % N == 0) return 1;
            double arg = -2 * Math.PI * k / N;
            return new Complex(Math.Cos(arg), Math.Sin(arg));
        }

        public static Complex[] fft(Complex[] x)
        {
            Complex[] X;
            int N = x.Length;
            if (N == 2)
```

```
            {
                X = new Complex[2];
                X[0] = x[0] + x[1];
                X[1] = x[0] - x[1];
            }
            else
            {
                Complex[] x_even = new Complex[N / 2];
                Complex[] x_odd = new Complex[N / 2];
                for (int i = 0; i < N / 2; i++)
                {
                    x_even[i] = x[2 * i];
                    x_odd[i] = x[2 * i + 1];
                }
                Complex[] X_even = fft(x_even);
                Complex[] X_odd = fft(x_odd);
                X = new Complex[N];
                for (int i = 0; i < N / 2; i++)
                {
                    X[i] = X_even[i] + w(i, N) * X_odd[i];
                    X[i + N / 2] = X_even[i] - w(i, N) * X_odd[i];
                }
            }
            return X;
        }
        public static Complex[] nfft(Complex[] X)
        {
            int N = X.Length;
            Complex[] X_n = new Complex[N];
            for (int i = 0; i < N / 2; i++)
            {
                X_n[i] = X[N / 2 + i];
                X_n[N / 2 + i] = X[i];
            }
            return X_n;
        }
    }
```

2.3 Вибір технологій для реалізації програмного комплексу

Для реалізації програмного додатку, а саме комплексу додатків, було обрано хмарне середовище Microsoft Azure. Структура Microsoft Azure являє собою сукупність віртуальних машин, що дозволяє користувачам

запускати універсальні віртуальні машини Microsoft Windows та Linux, а також налаштовувати образи цих систем для подальшої роботи.

Веб-сайти являють собою хостинг високої щільності веб-сайтів, які дозволяють розробникам створювати сайти за допомогою ASP.NET, PHP, Node.js або Python, і можуть бути розгорнуті за допомогою FTP, Git, Mercurial, Team Foundation Server або завантажені через портал користувача. Клієнти можуть створювати веб-сайти в PHP, ASP.NET, Node.js або Python або вибрати з декількох програм з відкритим кодом з галереї для розгортання. Це включає в себе один аспект платформи як послуги для платформи Microsoft Azure.

WebJobs - програми, які можуть бути розгорнуті в середовищі служби додатків для реалізації фонової обробки, яку можна викликати за розкладом, за вимогою або працювати безперервно. Служба Blob, Table і Queue може бути використана для спілкування між WebApps і WebJobs та надання доступу до команд.

Служби нотифікацій: SignalR сервіс який дозволяє передавати інформацію в режимі реального часу, що дозволяє сповіщавати користувачів.

Служби зберігання:

- Служби зберігання забезпечують API для REST та SDK для зберігання та доступу до даних у хмарному середовищі.
- Служба "Azure table" дозволяє програмам зберігати структурований текст у розподілених наборах суб'єктів, до яких можна звертатися по ключу розділу та первинному ключу. Це база даних без реляції, тобто NoSQL.
- Служба Blob дозволяє програмам зберігати неструктуровані текстові та бінарні дані у вигляді блоків, до яких можна отримати доступ через HTTP (S) шлях. Служба Blob також забезпечує механізми безпеки для контролю доступу до даних.

Служба черги дозволяє програмам спілкуватися асинхронними

повідомленнями, використовуючи черги.

Служба файлів дозволяє зберігати та отримувати доступ до даних у хмарі за допомогою API REST або SMB.

Управління даними:

- Azure Search надає текстовий пошук і підмножину структурованих фільтрів OData, використовуючи API REST або SDK.
- Cosmos DB - це служба бази даних NoSQL, яка реалізує підмножину оператора SQL SELECT у документах JSON. Redis Cache - керована реалізація Redis. StorSimple управляє завданнями зберігання між локальними пристроями та хмарним сховищем.
- SQL Database, раніше відома як SQL Azure Database, працює над створенням, масштабування та розширення додатків у хмарі за допомогою технології Microsoft SQL Server. Він також інтегрується з Active Directory та Microsoft System Center і Hadoop.
- SQL Data Warehouse - це служба зберігання даних, призначена для обробки обчислювальних та інтенсивних запитів даних на наборах даних, що перевищують 1 Тб.
- Azure Data Factory - це служба інтеграції даних, яка дозволяє створювати керовані даними робочі потоки в хмарі для оркестування та автоматизації руху даних та перетворення даних.
- Azure Data Lake - це масштабоване сховище даних та аналітичний сервіс для великих обсягів аналізу даних, які вимагають від розробників масових паралельних запитів.
- Azure HDInsight - це велика служба, яка використовує дані, яка використовує Hortonworks Hadoop на Microsoft Azure, і підтримує створення кластерів Hadoop з використанням Linux з Ubuntu.
- Azure Stream Analytics - бездротовий масштабований движок обробки подій, який дозволяє користувачам розробляти та запускати аналізи в режимі реального часу на кількох потоках даних із джерел, таких як пристрої, датчики, веб-сайти, соціальні

медіа.

Обмін повідомленнями

Сервісна шина Microsoft Azure ServiceBus дозволяє програмам, що працюють на Azure або на зовнішніх пристроях, спілкуватися з Azure. Це допомагає створювати масштабовані та надійні програми в сервіс-орієнтованої архітектурі (SOA). Microsoft Azure ServiceBus підтримує чотири різних типи механізмів зв'язку:

- Посередницькі вузли, які забезпечують появу подій і телеметрії у хмарі в масовому масштабі, з низькою затримкою та високою надійністю. Наприклад, концентратор подій може бути використаний для відстеження даних з мобільних телефонів, таких як координати місцезнаходження GPS в реальному часі.
- Черги, які дозволяють одностороннє спілкування. Програма відправника надсилатиме повідомлення до черги службової шини, і одержувач буде читати з черги. Хоча для черги може бути декілька читачів, лише один може обробляти одне повідомлення.
- Теми, які забезпечують одностороннє спілкування за допомогою шаблону підписки. Це схоже на чергу, однак кожен абонент отримає копію повідомлення, відправленого на тему. За бажанням, абонент може відфільтрувати повідомлення на основі конкретних критеріїв, визначених абонентом.
- Реле, що забезпечують двонаправлене спілкування. На відміну від черг та тем, реле не зберігає повідомлення в польоті у власній пам'яті. Замість цього, він просто передає їх до цільової програми.

Програми C # запускаються на платформі .NET Framework, що є невід'ємною частиною Windows, яка включає в себе віртуальну систему виконання, яка називається загальною мовою виконання (CLR) та уніфікованим набором бібліотек класів. CLR - це комерційна реалізація корпорацією Microsoft спільної мовної інфраструктури (CLI), міжнародного

стандарту, що є основою для створення середовища виконання та розробки, в якій мови та бібліотеки працюють разом.

Вихідний код, написаний в C #, складається на проміжну мову (IL), що відповідає специфікації CLI. Код IL та ресурси, такі як растрові зображення та рядки, зберігаються на диску у виконуваному файлі, який називається збіркою, зазвичай з розширенням .exe або .dll. Збірка містить маніфест, який надає інформацію про типи, версії, культуру та вимоги безпеки асамблеї[11].

Коли програма C # виконується, збірка завантажується в CLR, яка може виконувати різні дії на основі інформації в маніфесті. Тоді, якщо вимоги до безпеки відповідають вимогам, CLR виконує компіляцію в режимі реального часу (JIT), щоб перетворити IL-код на власні інструкції машини. CLR також надає інші послуги, пов'язані з автоматичною збіркою сміття, обробкою виключень та управління ресурсами. Код, який виконує CLR, іноді називають "керованим кодом", на відміну від "некерованого коду", який складається в нативній машинній мові, яка націлена на конкретну систему. Наступна діаграма ілюструє співвідношення часів і часу виконання файлів вихідного коду C #, бібліотек класів .NET Framework, збірок та CLR.

Висновки до розділу 2

Технічний аналіз є дуже важливою складовою маркетингового аналізу, який, в свою чергу, базується на вивченні графіків та коефіцієнтів обміну або ринку.

Дані, з якими працює технічний аналіз, відображає результати дій учасників обміну, які, у свою чергу, відображаються на цінових картах різних варіантів (свічки, чарти та ін.). Їх в свою чергу можна розглядати як функції, для яких будь-які математичні умови готові до застосування.

У цьому розділі було запропоновано інструменти для прогнозування руху цін за допомогою процесу згортки, в результаті чого ми можемо визнати поведінкові схеми в діаграмі руху цін у реальному часі. Також, знайшовши шаблони, можна порівняти ситуацію на ринку з тим, що трапилося раніше, що дозволяє зробити припущення щодо руху цін у майбутньому.

3. РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ АНАЛІТИКИ ТА ЗІСТАВЛЕННЯ ПРОГНОЗУ КОЛИВАНЬ КРИПТОВАЛЮТИ

Розроблювальна система представляє собою комплекс додатків, що призначені для збору, аналізу та зіставлення прогнозу по курсу криптовалюти. Основною задачею системи є те, що вона дозволяє користувачу відслідковувати процеси коливання курсу, та вчасно отримувати інформацію про зміну напрямку курсу. Також вона містить модуль моніторингу біржі на даний момент.

Для розробки мікро-сервісних систем було сформовано вимоги, щодо функціональних можливостей програмного забезпечення.

3.1 Вимоги до програмного забезпечення

3.1.1 Список термінів і визначень

Користувач – особа, яка аналізує процес коливання курсу криптовалюти, та займається трейдингом або ж інвестує в криптовалюту свої кошти.

Мікросервісна архітектура - також відома як архітектура мікросервісів - це архітектурний стиль, який структурує програму як сукупність поєднання вільно пов'язаних компонентів, які реалізують бізнес-можливості. Архітектура мікросервісу дозволяє безперервно доставляти / розгортати великі, складні програми. Це також дає змогу гнучко розвивати свій комплекс програм.

Алгоритм зіставлення прогнозу – низка задач, в якій вирішується обрахунок та аналіз графіку який подається, та відповідає даним, які пересилаються між підпрограмами.

Підпрограма – програма, яка реалізовує певну частину бізнес логіки, яка містить на вході необхідну множину вхідних даних та генерує на виході множину вихідних даних, які можуть пересилатися для виконання іншої підпрограми.

3.1.2 Загальні вимоги

Призначення розроблюваної системи – надання функціональних можливостей для аналізу та зіставлення прогнозу коливань курсу криптовалюти на певний період часу.

Система надає наступний спектр можливостей пов'язаних з:

- налаштуванням основних складових частин мікросервісної системи;
- автоматичним збором даних з криптовалютних бірж, які задають вхідні дані для аналізу та зіставлення прогнозу на заданий в конфігурації проміжок часу;
- виконанням обчислень в заданій мікросервісній системі з отриманими вхідними даними з обраної криптовалютної біржі;
- моніторингом стану курсу криптовалюти на обраній біржі та співвідношенням валют.

Простота інтерфейсу та гнучкість конфігурації мікросервісної системи дозволяють користувачу користуватися нею з самого початку, та для розробників має потенціал та простоту в розширенні проекту.

3.1.3 Вимоги користувача

Розроблювальна мікросервісна система надає наступні функціональні можливості для користувача:

- користувач може продивлятися актуальну інформацію про стан курсу з криптовалютної біржі;
- користувач може переглядати результати зіставлених прогнозів;
- користувач може підписатися на отримання повідомлень щодо прогнозів по зміни курсу;
- користувач може отримувати інформацію про стан системи в певний момент час виконання;
- користувач може продивлятися проміжні та кінцевий результат виконання обробки.

3.1.4 Специфікація функціональних вимог

1. Система повинна виконувати аналіз та зіставлення прогнозу напрямку курсу криптовалюти за певний проміжок часу. Відповідно до заданої архітектури мікросервісної системи та заданого алгоритму повинні бути виконані порівняння заданих сигналів зі збереженими шаблонами.

2. Система дає можливість змінювати налаштування щодо часу періоду зборів інформації та частоти обчислень які виконує додаток.

2.1. Система повинна надавати можливість переглядати стан курсу на даний момент. Після запуску процесу зіставлення прогнозу, користувач повинен мати змогу підписатися на повідомлення, щодо результату виконання обчислень.

2.2. Стан курсу та стан результатів щодо зіставлення прогнозів повинні обновлюватися в реальному часі без перезавантаження сторінки.

2.3. Всі зміни елементів системи повинні відображатися на інтерфейсі користувача.

3. Система повинна надавати можливість задавати налаштування підпрограм, які повинні виконуватися на підлеглих ресурсах мікросервісної архітектури.

4. Система повинна відображати стан даних системи в кожен момент часу.

5. Система повинна показувати інформацію про стан процесу модуля зіставлення прогнозів, а саме логи консольного додатку, що відображає процеси, які в ньому відбуваються в певний момент часу. Крім цього, система повинна надавати можливість записувати ці логи до текстового файлу.

6. Система повинна показувати інформацію про стан підлеглих модулів мікросервісної архітектури. Якщо модуль не виконує підпрограму, то система повинна показувати, що під час виконання щось пішло не так.

3.2 Архітектура системи

Загальна структура архітектури мікросервісної системи аналізу та зіставлення прогнозу криптовалютних коливань складається з наступних частин:

- інтерфейс користувача – модуль системи, через який користувач отримує інформацію про результат зіставлення прогнозу та актуальну інформацію про ринок на даний момент;
- консольний додаток – модуль, який безпосередньо аналізує та робить зіставлення прогнозу на певний встановлений в конфігурації проміжок часу;
- блок історії – модуль, який збирає інформацію про актуальний стан на ринку та надає цю інформацію користувачу до модулю інтерфейсу;
- блок роботи з шаблонами фігур – модуль, який працює з заздалегідь підготовленими шаблонами фігур які представлені у виді числового ряду;
- модуль нотифікацій – модуль, який будує та відправляє з наданої інформації. Також даний модуль створює дескриптори завдань та дескриптори даних, для тих даних, які користувач задав у файли з вхідними даними.

Архітектури системи наведено на рис. 20. Загалом вона складається з чотирьох слоїв, які взаємодіють між собою, в більшості випадків за допомогою обміну тексту в форматі JSON, який десеріалізується та серіалізується на різних етапах передачі даних.

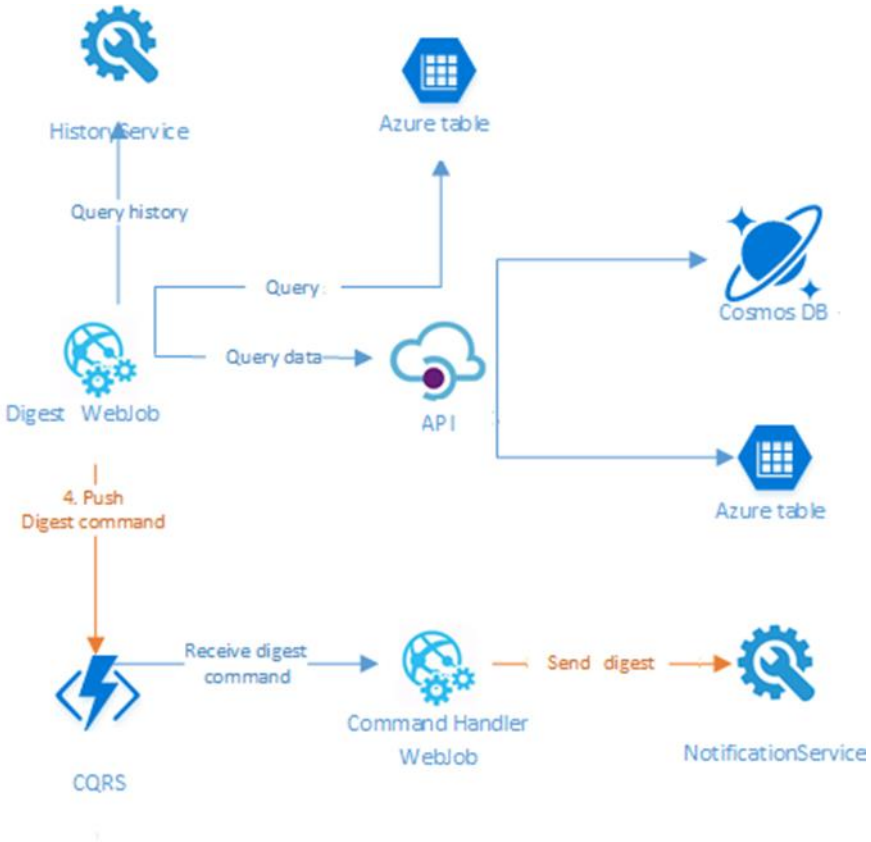

Рис. 19. Архітектура системи

Command Publisher – бібліотека для публікації команд, які містять в собі структуру JSON файлу. Команди містять в собі подію, тип та дані, які необхідні для опрацювання. У випадку даного додатку - це вхідні параметри, які містять в собі колекцію та налаштування системи.

Digest WebJobs – консольний додаток, який виконаний за допомогою бібліотек WebJob SDK 3.0 та виконує функції збору та синхронізації даних, також він побудований з набору функцій, які відповідають за свою функціональність, наприклад, за синхронізацію та збір даних відповідає одна функція; за обрахунок та зіставлення прогнозу – інша; також присутня міграція даних на інше сховище, в даному випадку, Cosmos DB.

History Service – набір сервісів, який дозволяє з’єднуватися з біржою, та отримувати дані які необхідні для моніторингу та зіставлення прогнозу.

Notification Service – набір сервісів, який дозволяє сповіщавати користувача про зміни в системі, та відправляти звіт чи повідомлення про

певну подію.

Digest WebJobs - консольний додаток, який виконаний за допомогою бібліотек WebJob SDK 3.0, та виконує функції розподілення та виконання команд, які знаходяться в черзі на виконання.

Багатошарова архітектура - один з найпопулярніших способів організації структури веб-додатків. У простій її варіації, виглядає як пошарова структура, додаток ділиться на 3 частини: шар даних, шар бізнес-логіки й шар призначеного для користувача інтерфейсу.

В шарі даних є певний Repository, який абстрагує логіку роботи, з певним типом бази даних.

В шарі бізнес-логіки є об'єкти, які інкапсулюють бізнес-правила (зазвичай їх назви варіюються в межах Services / BusinessRules / Managers / Helpers). Запити користувача проходять від UI крізь бізнес-правила, і далі через Repository проводиться робота зі сховищем даних.

З такою архітектурою запити на отримання і зміна даних, як правило, проводяться в одному і тому ж місці - в шарі бізнес-логіки. Це досить простий і звичний спосіб організації коду, і така модель може підійти для більшості додатків, якщо в цих додатках кількість користувачів з часом значно не змінюється, додаток не відчуває великих навантажень і не вимагає значного розширення функціоналу

Але якщо веб-ресурс стає досить популярним, може стати питання про те, що одного сервера для нього недостатньо. І тоді постає питання про розподіл навантаження між декількома серверами. Найпростіший варіант швидко розподілити навантаження - використовувати декілька копій ресурсу і реплікацію бази даних. А враховуючи, що всі дії такої системи ніяк не розділені, це породжує нові проблеми.

Класична багатошарова архітектура (рис. 21) не забезпечує легкого вирішення подібних проблем. Тому непогано було б використовувати підходи, в яких ці проблеми вирішені з самого початку. Одним з таких підходів є CQRS (рис. 22).

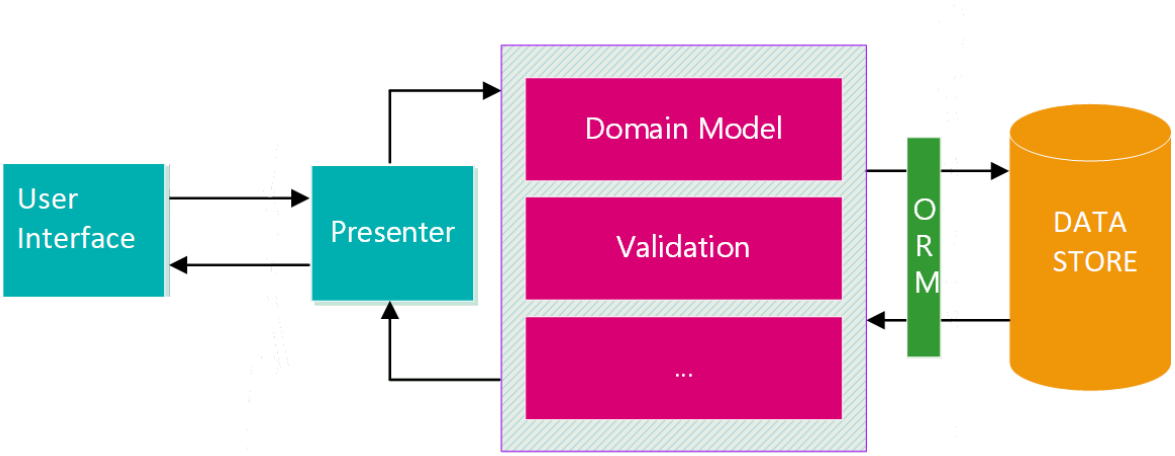

Рис. 20. Структура системи без CQRS підходу

CQRS - підхід проектування програмного забезпечення, при якому код, що змінює стан, відділяється від коду, який просто читає цей стан. Подібне розділення може бути логічним і гуртуватися на різних рівнях. Крім того, воно може бути фізичним і включати різні ланки, або рівні.

В основі цього підходу лежить принцип Command-query separation(CQS). Основна ідея CQS в тому, що в об'єкті методи можуть бути двох типів:

- Queries (запити): методи повертають результат, не змінюючи стан об'єкта. Іншими словами, у Query не існує будь яких побічних ефектів;
- Commands (команди): методи змінюють стан об'єкта, не повертаючи значення.

У CQS Query є одна особливість. Раз Query ніяк не змінює стан об'єкта, то методи типу Query можна добре розпаралелити, розділення припадає на операції читання. Якщо CQS оперує методами, то CQRS піднімається на рівень об'єктів. Для зміни стану системи створюється клас Command, а для вибірки даних - клас Query. Таким чином, ми отримуємо набір об'єктів, які змінюють стан системи, і набір об'єктів, які повертають дані.

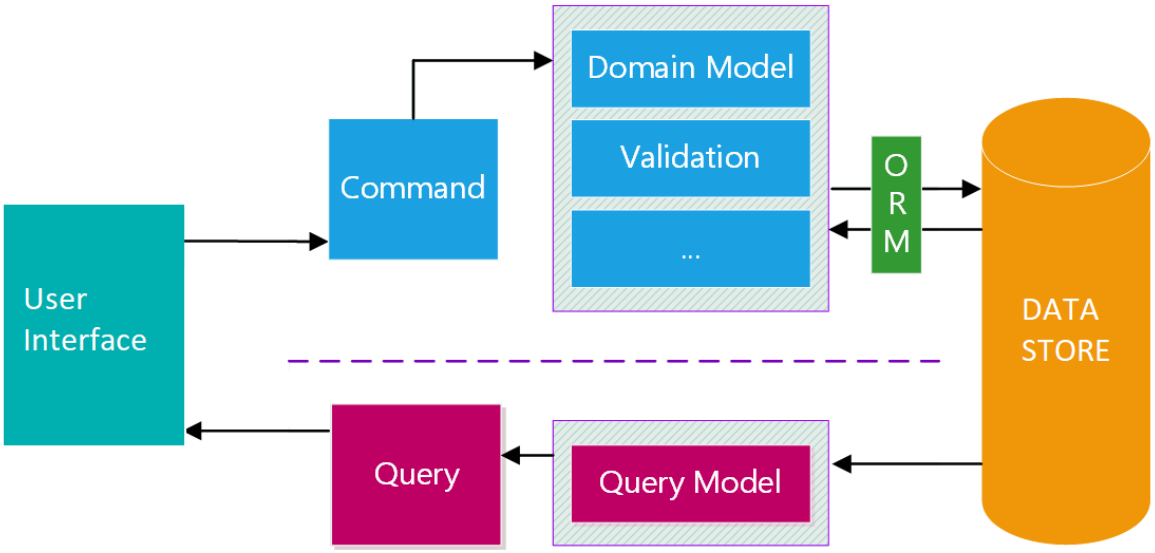

Рис. 21. Структура системи з CQRS підходом

CQRS, досягається групуванням операцій запиту в одному рівні, а команд - в іншому. Кожен рівень має свою модель даних, свій набір сервісів і створюється із застосуванням своєї комбінації шаблонів і технологій. Ще важливіше, що ці два рівня можуть перебувати навіть в двох різних ланках і оптимізуватися окремо, ніяк не зачіпаючи один одного.

Просте розуміння того, що команди і запити є різними речами, має глибокий вплив на архітектуру ПО.

Наприклад, раптом стає легше передбачити і кодувати кожен рівень предметної області. Рівень предметної області в стеку команд потребує лише в даних, бізнес-правилах і правилах безпеки для виконання завдань. З іншого боку, рівень предметної області в стеці запитів може бути не складніше прямого SQL-запиту.

CQRS дозволяє оптимізувати конвеєри команд (рис. 22) і запитів будь-яким способом. При цьому оптимізація одного конвеєра не порушить роботу іншого. У самій базовій формі CQRS використовується одна загальна база даних і викликаються різні модулі для операцій читання і запису з прикладного рівня.

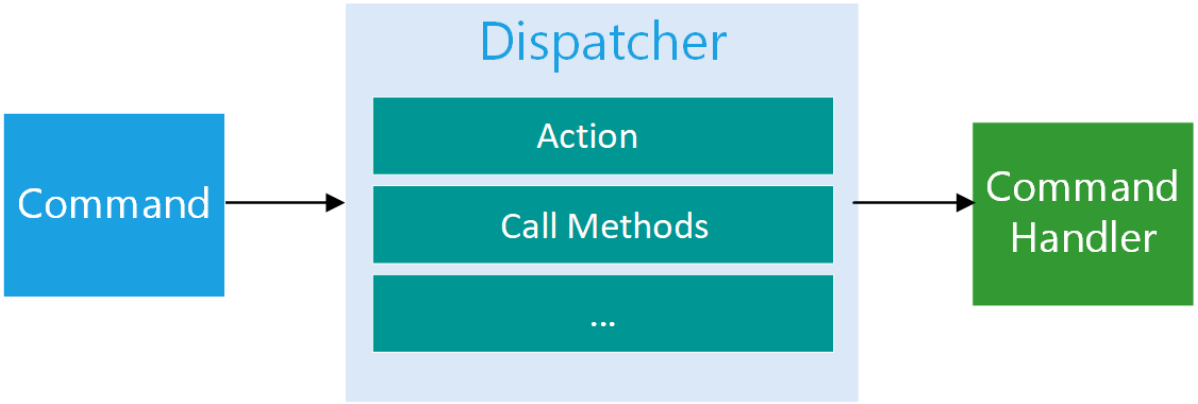

Рис. 22. Структура системи з CQRS підходом

3.3 База даних

Для розробки комплексу додатків було використано два основних сховища даних. Таких як Azure Table та Cosmos DB.

Azure Cosmos DB з самого початку розроблявся для глобального розподілу і горизонтального масштабування. Ця база даних пропонує готовий глобальний розподіл в будь-якій кількості регіонів Azure шляхом прозорого масштабування і реплікації даних незалежно від місцезнаходження користувачів. Забезпечує еластичне масштабування операцій запису і читання по всьому світу.

При цьому платити потрібно тільки за ті ресурси, які вам потрібні. Azure Cosmos DB забезпечує вбудовану підтримку для NoSQL і API OSS, включаючи MongoDB, Cassandra, Gremlin і SQL, а також пропонує кілька чітко визначених моделей узгодженості.

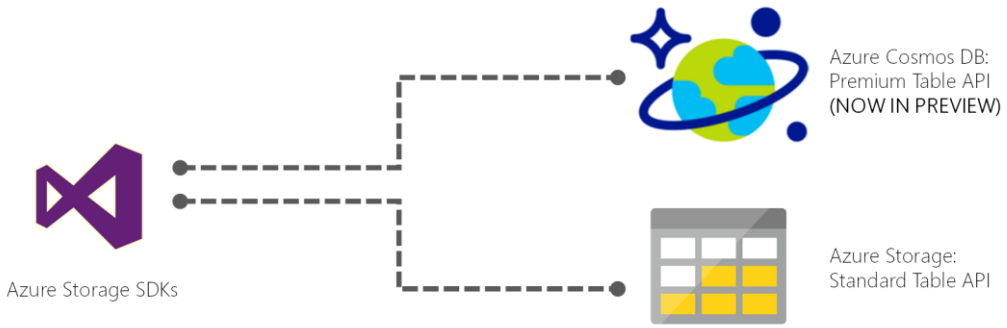

Рис. 23. Структура системи з CQRS підходом

3.4 Висновки до розділу 3

У даному розділі було описано розроблений програмний комплекс. Докладно описано кожен модуль який використовується в системі, та за яку саме діяльність він відповідає. Наведено діаграми архітектури проекту та скріншоти візуальної частини програмної реалізації. Продемонстровано можливості конфігурації додатків. Архітектура базується на принципах CQRS шаблону проектування, та базується на мікросервісній архітектурі Microsoft Azure. Для роботи всієї системи, повинні бути налаштовані всі окремі сервіси, що є як і плюсом такого підходу, так і недоліком. Оскільки, для налаштування всього проекту з нуля, потрібно провести багато часу за конфігурацію окремо діючих модулів. Які в свою чергу відповідають за різну функціональність проекту.

Розроблений програмний комплекс надає зручний спосіб для моніторингу та перегляду зіставленого прогнозу на певний період часу. Завдяки розробленому веб додатку, який відображає дані, що були отримані з біржі, та в наслідку оброблені розробленим алгоритмом.

4. АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ АНАЛІТИКИ ТА ЗІСТАВЛЕННЯ ПРОГНОЗУ КОЛИВАНЬ КРИПТОВАЛЮТИ

Розроблена система представляє собою комплекс додатків, що призначені для збору, аналізу та зіставлення прогнозу по курсу криптовалюти. Також система містить модуль моніторингу біржі, яка надає представлення про ринок на даний момент часу. Далі буде проведено порівняння роботи розробленого додатку зі знайденим аналогом побудованим на нейронній мережі.

4.1. Порівняльний аналіз ефективності існуючих рішень з розробленою системою

Для порівняння було обрано існуюче рішення для зіставлення прогнозу на основі нейронної мережі.

Можливість навчання – це одна з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Здібності нейронної мережі до прогнозування безпосередньо випливають з її здатності до узагальнення і виділення прихованих залежностей між вхідними та вихідними даними. Після навчання мережа здатна передбачити майбутнє значення якоїсь послідовності на основі декількох попередніх значень або якихось існуючих зараз чинників. Слід зазначити, що прогнозування можливо тільки тоді, коли попередні зміни дійсно в якійсь мірі визначають майбутні. Наприклад, прогнозування котирувань акцій на основі котирувань за минулий тиждень може виявитися успішним, тоді як прогнозування результатів завтрашньої лотереї на основі даних за останні 50 років майже напевно не дасть ніяких результатів.

Розглянемо на практиці застосування методу прогнозування за допомогою нейронних мереж. Для прикладу візьмемо дані з біржі Bitterx в

період з 01.11.2018 по 28.11.2018. Завдання полягає в тому, що на основі представленої статистичної інформації необхідно зробити прогноз на 3 дні. Як видно з графіка (рис. 22), з 13.11.2018 по 27.11.2018 відбулося падіння курсу. Далі проведемо аналіз ймовірностей, які були отримані при обрахунках нейромережею та власним програмним комплексом.

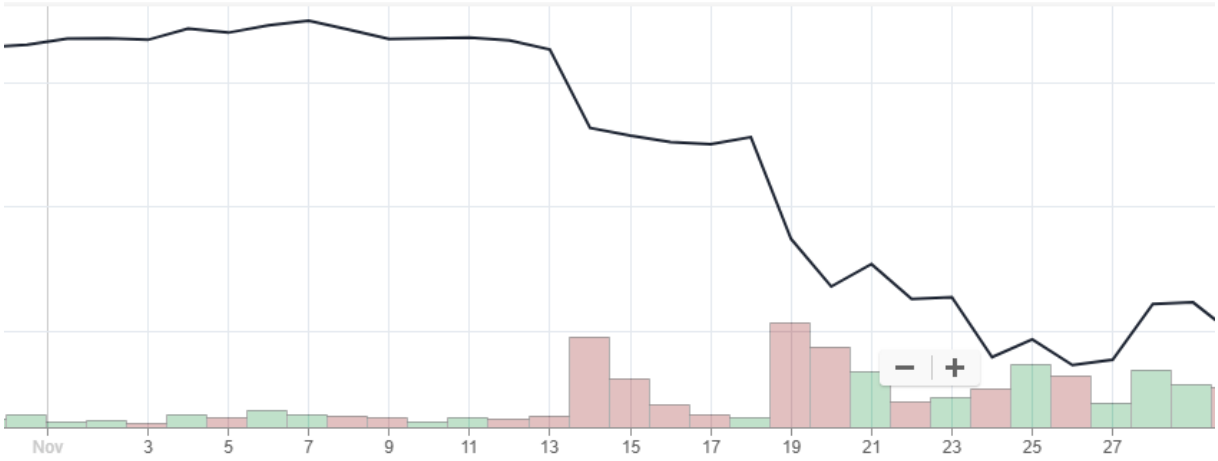

Рис. 22. Графік відношення USD-BITCOIN

Як і передбачалося, нейронна мережа видає достатній результат. Але навіть, незважаючи на всі видимі позитивні якості нейронних мереж не варто вважати їх якоюсь панацеєю. По-перше, нейронні мережі є чорним ящиком, який не дозволяє в явному вигляді визначити вид залежностей між членами ряду. Таким чином, конкретну нейронну мережу можна навчити будувати прогноз лише на строго фіксовану кількість кроків вперед (яку ми вказуємо в специфікації цієї мережі), отже, має місце сильна залежність від виду завдання[12,13]. По-друге, при наявності явної лінійності, простоти структури в задачі, здатність нейронних мереж до узагальнення виявляється слабшою по відношенню до класичних методів. Пояснюється це як раз нелінійністю мереж за своєю суттю. В таблиці 1 наведено порівняння точності результатів з нейромережею. Який був зіставлений, на основі даних з біржі Bittrex, за описаний вище проміжок часу.

Таблиця 1. Порівняння точності прогнозу з нейромережею

Дата	Нейромережа	Розроблений комплекс
01.11.18	✓	✓
02.11.18	✓	✓
03.11.18	✓	✓
04.11.18	✓	✓
05.11.18	✓	✓
06.11.18	✓	✓
07.11.18	✓	✓
08.11.18	✓	✓
09.11.18	✓	✓
10.11.18	✕	✓
11.11.18	✓	✓
12.11.18	✓	✓
13.11.18	✓	✓
14.11.18	✕	✕
15.11.18	✕	✓
16.11.18	✓	✓
17.11.18	✓	✓
18.11.18	✓	✓
19.11.18	✕	✕
20.11.18	✕	✕
21.11.18	✓	✓
22.11.18	✓	✓
23.11.18	✓	✓
24.11.18	✓	✓
25.11.18	✓	✓
26.11.18	✓	✓
27.11.18	✕	✓
28.11.18	✕	✓
В середньому	75%	85%

4.2. Аналіз ефективності розробленого програмного комплексу для аналітики та зіставлення прогнозу коливань криптовалюти з урахування особливостей роботи та окремих компонентів

Реалізований комплекс додатків складається з чотирьох основних модулів, а саме:

- Модуль нотифікацій;

Модуль який відповідає за сповіщення користувача в системі, про зміни, якщо після обробки зібраної інформації був зіставлений прогноз в якому визначено можливий знак про зміну тренду.

- Модуль Обробки та зіставлення прогнозу;

Модуль який відповідає за обробку даних які записуються до системи з криптовалютної біржі. На основі яких формується прогноз та записується в інше середовище, яке в свою чергу зберігає дані про прогноз на певну дату.

- Модуль збору даних з криптовалютних бірж;

Модуль який відповідає за збір даних з криптовалютної біржі. На основі яких формується прогноз та записується в інше середовище. Також сюди входить підпрограма яка дозволяє синхронізуватися з базою та видалити непотрібні дані. Але взагалі дані з системи не видаляються, а кладуться в архів, щоб в подальшому мати змогу працювати з величезними об’ємами даних, які були збережені. Також основною з причин створення подібного архіву стало те, що криптовалютні біржі не надають інформації на великий проміжок часу. А переважно оперують лише даними на місяць.

- Модуль веб застосунку.

Модуль який являє собою Azure Web Application, що виступає в ролі веб застосунку, яким користуються кінцеві користувачі продукту. В собі має модулі по відображені чарту на даний момент, та дані щодо доступних прогнозів на встановлений проміжок часу. Структура роботи даних побудована на принципі Redux, який наполягає на тому, що в системі повинен бути лише одне джерело даних, та дозволяє уникнути залежностей компонентів додатку від прямих включень сервісів, які оперують з API[14].

На рис. 16 зображено детальну схему взаємодії розробленого комплексу зі сховищем даних та іншими компонентами системи.

Клієнтський веб застосунок (Client App) представляє собою Azure Web Application, що виступає в ролі користувацького інтерфейсу, яким користуються кінцеві користувачі продукту.

Веб API застосунок (Data API) представляє собою сервіс, що виступає в ролі джерела для отримання даних, через яке можна витягти всі необхідні дані для відображення. Для отримання даних, потрібно зробити запит на конкретний метод в API, через HTTP(s) запит, та в результаті обробки запиту, буде повернено дані в текстовому форматі JSON. Який є стандартом для взаємодії додатків.

SQL database (sql інтерфейс до cosmosdb), який дозволяє працювати з нереляційною базою даних, за допомогою запитів звичайної реляційної бази SQL.

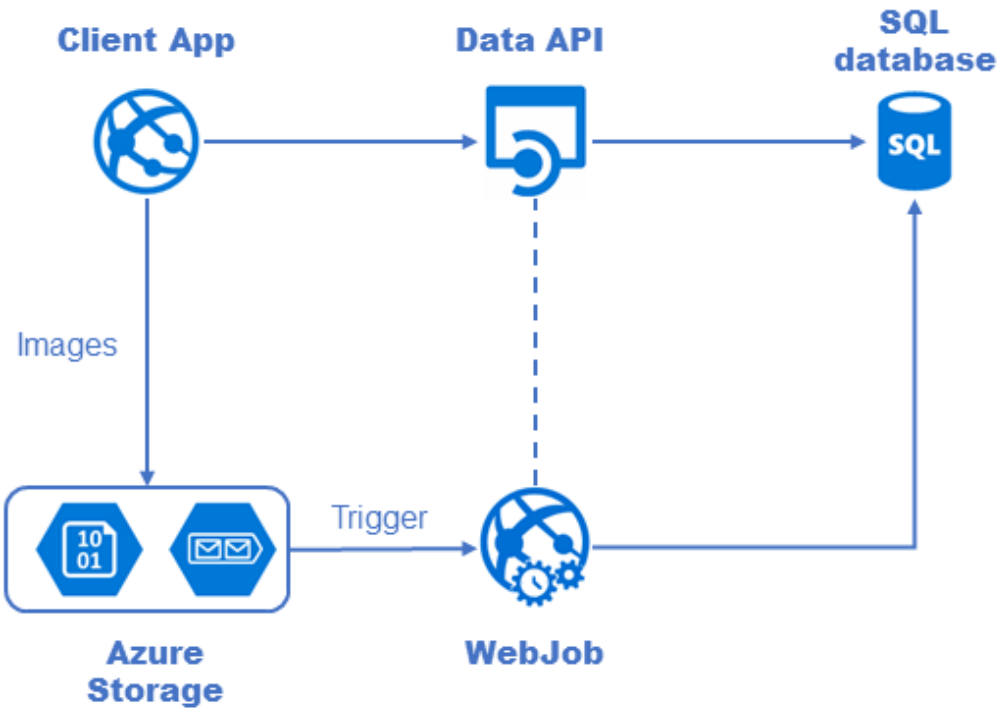

Рис 24. Схема взаємодії компонентів розробленої системи

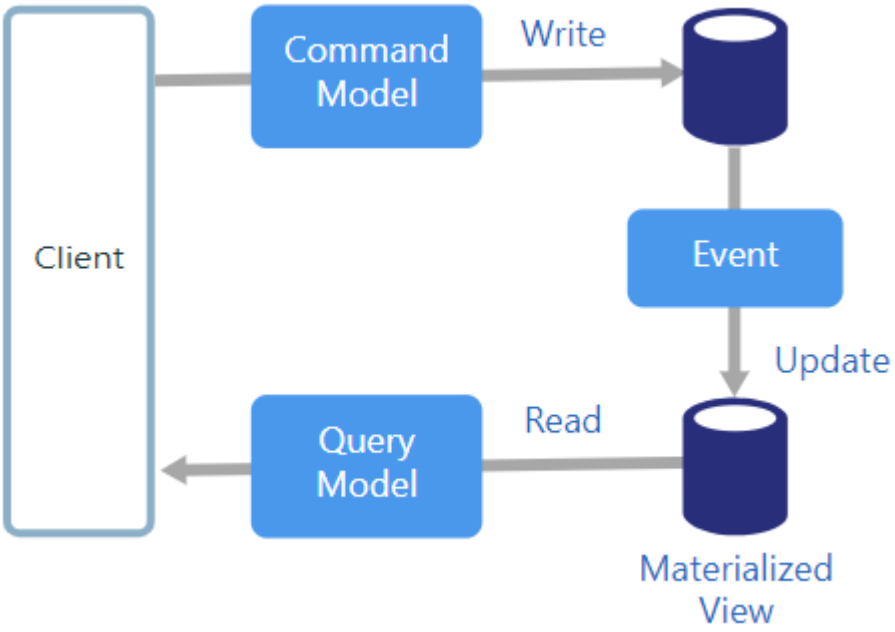

Рис 25. Схема роботи з командами

На (рис. 16) зображено схему взаємодії додатку зі сховищем команд. В основі роботи додатку, лежить принцип CQRS, який полягає в наступному:

Для виконання дії в системі, потрібно створити команду. Далі, ця команда буде збережена до сховища, в даному випадку Azure Table. Для обробки команди, повинен запрацювати тригер на нову команду, після чого, вона буде обрана в виконання, в залежності від типу команди, буде обрано відповідний блок інструкцій. Також, основним принципом є те, що при створенні команди, модуль не отримує нічого зайвого, та не тримає певний потік, в якому виконується підпрограма яку він повинен виконати.

В цьому аспекті, також є і недолік, який полягає в налаштуванні та впровадженні такого ПЗ, підтримці його в працездатному стані, моніторингу сервісів, але цей недолік можна вирішити за допомогою супутніх інструментів які присутні в хмарному середовищі Azure. Яке має масштабний та різнобічний набір інструментів для розробки та впровадження програмного забезпечення різних типів, починаючи від простих веб-додатків, закінчуючи складними багатошаровими системами.

CryptocurrencyForecasting.WebUI

USD-BTC forecast

Date	Status
2018-11-04	Stay
2018-11-05	Stay
2018-11-06	Stay
2018-11-07	Stay
2018-11-08	Stay
2018-11-09	Stay
2018-11-10	Stay
2018-11-11	Stay
2018-11-12	Stay
2018-11-13	Stay
2018-11-14	Stay
2018-11-15	Stay
2018-11-17	Stay
2018-11-18	Down
2018-11-19	Stay
2018-11-23	Down
2018-11-24	Stay
2018-11-28	Stay
2018-11-29	Stay
2018-12-01	Down
2018-12-02	Down

Рис 26. Скріншот роботи веб-додатку

4.3 Аналіз швидкодії та відмовостійкості системи

Система аналізу та зіставлення прогнозу коливань курсу криптовалюти, має працювати швидко, та мати високу відмовостійкість. Під швидкодією, розуміється швидка обробка даних (тобто відпрацювання алгоритму вчасно) для надання актуальної інформації. В той же час подібна система має працювати без переривань та відмов, тому, що це в свою чергу призведе до спотворення вихідної інформації, та до невчасного сповіщення користувачів про зміни курсів, що зведе сенс роботи до нуля. Тому для реалізації способу зіставлення прогнозу, був використаний математичний метод, який виконується в проміжку певного часу, який залежить від довжини послідовності, яка обробляється.

Для досягнення максимальної відмовостійкості розроблюваної системи, було обрано хмарне середовище Microsoft Azure, в якому можливо розподілити модулі системи на різних віртуальних образах операційних систем, що дозволяє розподілити навантаження.

За допомогою шаблону проектування CQRS, вдалося досягти незалежності компонентів одного від іншого, що дало значний приріст по швидкодії та відмовостійкості. Оскільки модулі не чикають один одного в процесі взаємодії, та спілкуються лише через чергу команд, яка являє собою централізоване джерело інструкцій для модулів. Далі в таблиці 5 було проведено порівняння швидкодії розробленого комплексу з нейромережою.

З якої видно, що по швидкодії нейромережі програють, так як вони є доволі громіздкими комплексами та час їх виконання залежить від кількості нейронів, які були створені при навчанні. Також все залежить від того, які дані надходять для опрацювання, таким чином, можуть виконатися не всі існуючі зв’язки нейронів, що дасть більшу швидкість. Над відміну від нейромережі, розроблений спосіб працює завжди за одним і тим самим алгоритмом, що дає змогу визначити тривалість процесу обрахунку, який залежить лише від об’єму наданих даних та потужності платформи на якій він виконується.

Таблиця. 2 Порівняння швидкодії з нейромережею

Дата	Нейромережа	Розроблений комплекс
01.11.18	1702	339
02.11.18	1399	439
03.11.18	1169	383
04.11.18	1509	335
05.11.18	1987	316
06.11.18	1529	385
07.11.18	1429	493
08.11.18	1848	412
09.11.18	1816	480
10.11.18	2270	224
11.11.18	1684	356
12.11.18	2903	344
13.11.18	1184	497
14.11.18	1640	494
15.11.18	2211	388
16.11.18	1073	428
17.11.18	2320	487
18.11.18	2007	369
19.11.18	2111	304
20.11.18	1003	267
21.11.18	2090	481
22.11.18	2382	383
23.11.18	2446	230
24.11.18	1548	379
25.11.18	1285	398
26.11.18	2370	306
27.11.18	1356	429
28.11.18	1168	359
В середньому	1768	383

Висновки до розділу 4

У четвертому розділі було порівняно різні аспекти системи з існуючими методами, а саме обраної нейромережі. Проведено порівняння швидкодії та відмовостійкості. В результаті яких виявилося, що розроблений спосіб перевершує результати нейронної мережі на проміжках часу, в яких відбуваються значні коливання та зміни трендів напрямку ринку. Так як розроблений спосіб працює з фігурами зміну тренду, які краще підходять саме для таких випадків. Щоб досягти подібних результатів за допомогою нейромережі, потрібно провести багато часу за її навчанням та підбором ваг нейронів. Що є досить складною та комплексною роботою, в альтернативу від розробленого способу, який працює завжди з заздалегідь підготовленими шаблонами, та в результаті, дає позитивний результат.

5. СТАРТАП СКЛАДОВА ЧАСТИНА ПРОЕКТУ

5.1. Опис проблеми та дерево проблем.

5.1.1. Анотація проекту.

Проект спрямований на розробку та тестування методу аналізу та прогнозування коливання криптовалют. Розробка передбачає винайдення методу ефективного аналізу та прогнозування коливання валют на різних біржах криптовалют. Тестування передбачає аналіз кореляції прогнозованих та реальних даних, програмними засобами для подальшого використання розробленого методу в межах даної системи.

5.1.2. Опис проблеми

Розробка методу підвищення ефективності роботи трейдерів криптовалютних бірж та копаній які займаються трейдингом, обумовлена тим, що кількість бірж з кожним днем росте, інформація оброблюється та перезаписується в рамках кожної з бірж, так з’явилися ідея про створення єдиного сервісу для обробки та аналізу існуючих бірж, методом динамічного збору інформації та обробки в рамках системи.

На даний час подібні рішення досягаються створенням власних алгоритмів, які основані на зміні направлення курсу. Проте, як вже було сказано вище, системи, що надає подібний функціонал, немає, адже для цього необхідно враховувати типи систем їх особливості та об’єми інформації з якою відбуваються дії. Також є варіант статичного розпаралелювання задач на стадії розробки, що виводить систему на максимальну швидкість, проте це можливо лише у випадку, якщо розробник знає та описує у програмному коді всі можливі варіанти розвитку подій за допомогою потоків програмних засобів.

Основна спрямованість проекту – винайдення методу, що дозволить прискорити розподіл та аналіз даних з бірж. Прискорення має відбуватися за рахунок генерації наступної задачі на етапі виконання поточної

(попередньої). Використання даного методу дозволить проводити кореляцію з реальними показниками та оперяючись на справджуваність робити прогнози.

5.1.3 Мета та завдання проекту відповідно до проблеми.

Метою проекту є винайдення та тестування системи для прискорення роботи по аналізу та прогнозування курсу криптовалют. Для цього необхідно виділити вже існуючі методи підвищення ефективності, виявити їх недоліки та переваги, та обрати шлях розвитку проекту відповідно до виявлених переваг та недоліків. Також завданням проекту є створення модулю для отримання статистичної інформації для демонстрування та порівняння з аналогами винайденого методу. Відповідно презентувати отриманні результати, та знайти інвестиції для встановлення подібних систем чи заміни існуючих які є менш оптимальними. Залучити сторонніх розробників методом подальшого випуску вільного пакету програмних засобів, що дозволять розробникам створювати власні методи чи ПЗ на основі розробленого методу, що дасть можливість використовувати та впроваджувати метод в інші системи.

5.1.4 Дерево проблем.

Дерево проблем проекту зображено на рисунку 27.

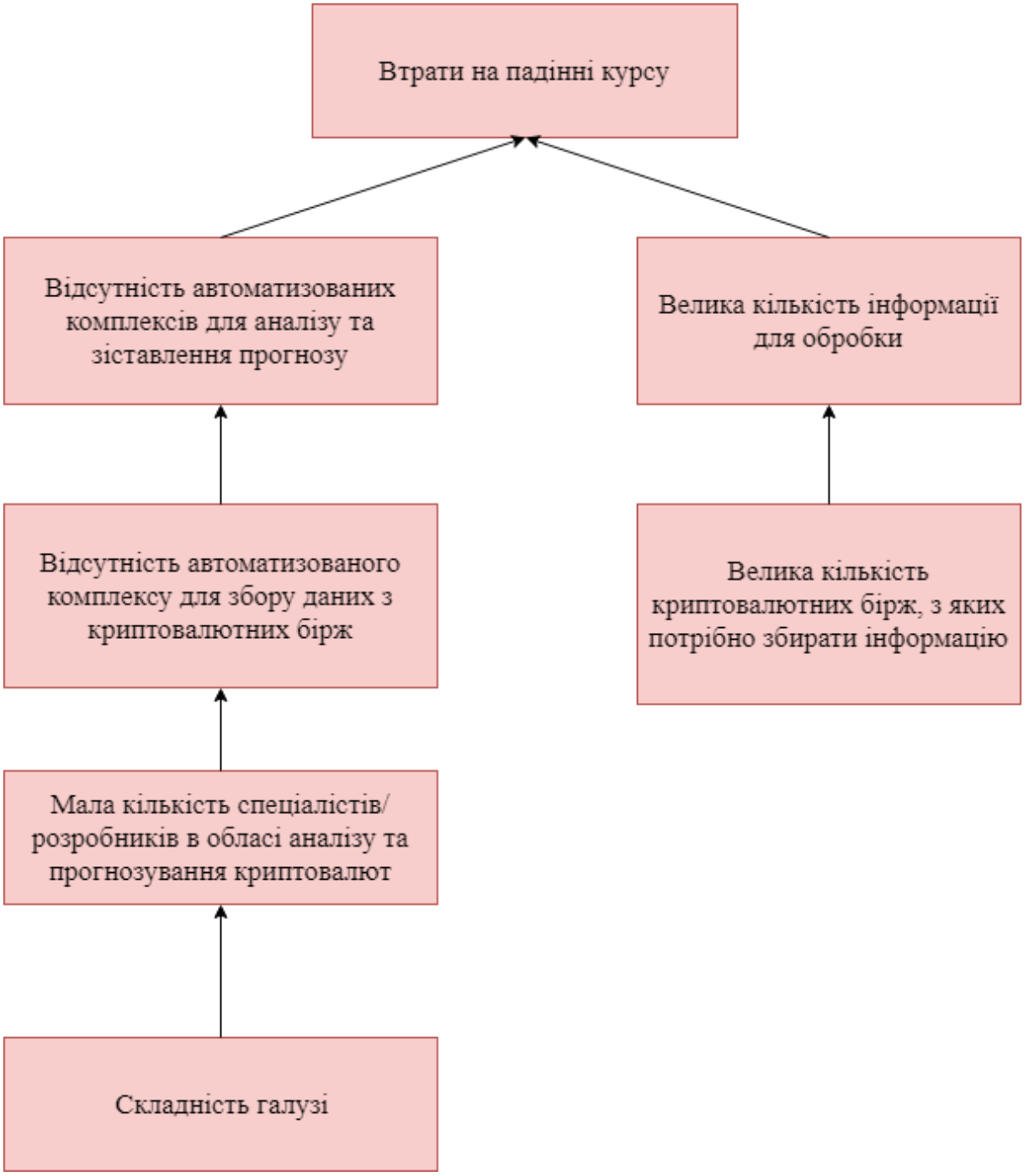

Рис. 27. Дерево проблем проекту.

5.2. Аналіз зацікавлених сторін проекту.

Зацікавлені сторони проекту та оцінка їх важливості та зацікавленості зібрані у таблиці 3.

Таблиця 3. Зацікавлені сторони проекту

Група зацікавлених осіб	Інтереси групи в проекті	Умови довгого співробітництва з проектом	Важливість	Зацікавленість
Внутрішні зацікавлені сторони проекту				
Розробник проекту	Виконання проекту; Досягнення цільових показників проекту	Подальший розвиток проекту; Притік інвестицій	Висока (10)	Висока (10)
Команда керування проектом	Досягнення цільових показників проекту; Подальший розвиток технологій проекту	Подальший розвиток проекту; Притік інвестицій; Збільшення особистого доходу	Висока (9)	Висока (9)
Інвестори проекту	Отримання зазначених доходів від участі в проекті; Подальший розвиток проекту; Досягнення цілей	Збільшення прибутку від інвестицій; Вигідніші умови підтримки проекту	Висока (9)	Висока (9)

Внутрішньо – корпоративні зацікавлені сторони проекту				
Менеджмент проекту	Розвиток проекту; Збереження притоку робочих місць;	Приріст робочих місць; Можливість кар’єрного росту; Реклама	Висока (9)	Середня (6)
Акціонери	Приріст доходності; Ріст загальної вартості проекту	Збільшення вартості проекту	Висока (9)	Висока (9)
Побічні співробітники	Кар’єрний ріст	Кар’єрний ріст	Середня (6)	Висока (9)
Зовнішні зацікавлені сторони проекту				
Копанії чи фіз. особи	Використання продукту проекту для прискорення роботи; Якість власного продукту; Власний технологічний розвиток	Розвиток технологічного фону	Нижче середнього (4)	Висока (9)
Наукові дослідницькі центри / заклади	Прискорення процесу досліджень	Підтримка продукту	Нижче середнього (4)	Висока (9)

Побічні зацікавлені сторони проекту				
Розробники ПО	Використання потокових технологій	Оновлення технологій	Низька (3)	Низька (3)

5.2.2 Аналіз зацікавлених сторін проекту.

Як видно з таблиці 3, зацікавлені сторони були розділені на 4 основні групи.

1.Внутрішні зацікавлені сторони проекту.

Тут зібрані особи, що напряму зацікавлені у проекті та його подальшому розвитку та встановленню на підприємствах закладах тощо, серед яких інвестори, для яких важливо мати прибуток з проекту, розробник проекту, який безпосередньо є керівником усього, що відбувається, як всередині команди так і зовні, команда проекту, що слідкує за виконанням проекту.

2. Внутрішньо – корпоративні зацікавлені сторони проекту

Ця група має менш серйозне відношення до проекту, проте без них його існування та подальший розвиток неможливі, серед них майбутні власники акцій, до яких входять і особи з першої групи, тому їх важливість та зацікавленість тісно перетинається з учасниками першої групи, а також до акціонерів належать побічні особи, зацікавленість яких може бути менша.

Побічні співробітники - їх важливість не така висока, особливо на перших етапах розвитку, - по перше, їх кількість невелика, по друге, вони не несуть такої сильної важливості, адже не пов’язані напряму з запуском, чи впровадженням розроблюваного методу у використання, проте, якщо

проект отримає подальший розвиток, їх кількість зросте, а співробітники, що були присутні на старті, отримують більші доходи.

3. Зовнішні зацікавлені особи.

Це всі хто зацікавлений саме у отриманні доступу до використання методу, тобто «цільова аудиторія» методу, до неї входять усі фірми, підприємства, корпорації, заклади, що працюють з інформацією у тому, чи іншому вигляді, а також можуть бути носіями статистичних даних, або тестовими платформами методу.

4. Побічні зацікавлені сторони.

Усупереч тому, що дана група має низьку зацікавленість та невисоку важливість, є потужним рушієм для популяризації методу, адже незалежні програмісти, можуть використовувати метод для створення власних продуктів у майбутньому, що принесе його у постійне використання.

5.3 Опис наукового проекту та технології.

Науковим продуктом у рамках наукової дисертації є спосіб аналізу та прогнозування коливань курсу криптовалют, що і є науковою назвою продукту.

Даний метод призначений для використання будь-якими закладами, які працюють з криптовалютами, чи працюють з нею. Даний метод спростить закладам процес роботи з інформацією, шляхом аналізу та впорядкування інформації з різних бірж. Також для цього можна використати графіки.

На ранніх етапах розробки тестування подібного методу буде проводитись методом збору статистичних даних відносно швидкодії, за допомогою роботи дата-центру з поточною обробкою даних, наприклад, обрахування чи зберігання, даний модуль буде створено власноруч засобами написання програмного коду. В наступних етапах застосунку

методу на підприємствах тощо, дані підприємства будуть слугувати так званою платформою для тестування та збору інформації для подальшого вдосконалення методу. Таким чином розвиток методу ніколи не припиниться.

Для реалізації методу на абстрактному рівні виділено декілька стадій:

- Аналіз різних підходів до аналізу та прогнозування курсу криптовалют та виділення їх переваг та недоліків;
- Створення систем на основі виробленого способу;
- Тестування написаної системи порівняно з існуючими методами;
- Написання самої системи;
- Збір статистичних даних та порівняння їх з реальною статистикою;
- Усунення недоліків, якщо це можливо, базуючись на отриманих даних з тестової платформи.

5.4 Бізнес рішення та основні характеристики бізнес - продукту.

5.4.1 Резюме продукту

Результатом науково – дослідницької роботи буде ПЗ та документація до нього. Дане ПЗ допоможе прискорити роботу трейдерів або компаній, які займаються криптовалютою та її аналізом. Далі буде розглянуто принцип розробки, користь продукту, способи підтримки та подальший розвиток, що дасть абстрактне розуміння про продукт, що буде отриманий в результаті дослідницької роботи. Також далі будуть розглянуті можливості та шляхи вводу ПЗ на виробництва різних типів, а також можливості отримання ПЗ та подальшого використання у «домашніх умовах», також будуть описані необхідні вимоги для встановлення ПЗ.

5.4.2 Опис продукту

Результатом дослідницької роботи буде ПЗ, але слід зазначити, що ПЗ далеко не основний виробничий продукт. В основі ПЗ лежить метод збору інформації кореляції курсу, тобто результатом роботи буде саме метод, який можна встановлювати або інтегрувати в різні систем, що працюють з великими об’ємами інформації.

Основним плюсом при використанні продукту є прискорення обробки інформації та збірок різноманітних програмних продуктів, такий метод може знадобитися: трейдерам, компаніям, які працюють з криптовалютою.

Користувачам які володіють криптовалютою, розробникам забезпечення.

Основною особливістю розробки є простота в інтегруванні, тобто даний метод просто реалізований у вигляді додатку або бібліотеки, що підключається до побічного проекту або інтегрується в процес, що дозволяє розподіляти навантаження методом поточних обчислень.

Процес «продажу» методу буде відбуватися методом інтегрування методу в окремий проект або цілком на виробництво або підприємство, та

подальша його підтримка за оплату послуг по підтримці, щомісяця та за використання технології. У випадку з розробниками власних проектів їм буде надана скорочена користувацька бібліотека, що проста в інтеграції, проте не підходить для великих проектів. Для отримання доступу до технології необхідно замовити інтегрування в проект та обов'язково подальшу підтримку, до якої буде прив'язаний окремий спеціаліст.

Продукт знаходиться на стадії постійного розвитку, що дасть можливість постійного доповнення методу та пристосування його до різних типів даних. Це є однією з проблем оптимізації роботи трейдингових операцій, адже різні типи інформації та різні системи виконують обробку та зберігають по-різному. Бізнес рішенням є постійна підтримка та постійній розвиток саме методу, адже ПЗ буде встановлюватись саме на момент дійсної версії методу, тому за оновлення та підтримку необхідне додаткове фінансування команди підтримки та безпосередньо за отримання нової версії продукту. Таким чином, оскільки результатом роботи, окрім безпосереднього прискорення є статистичні дані кожного власника системи, система обробки інформації буде використовувати власний спосіб зберігання і прискорення, тому затримок у оновленні даних буде менше, що дасть можливість своєчасно та оперативно помічати будь-які зміни у системах, що обслуговуються.

В майбутньому метод перейде на інтегрування на великі серверні частини, що значно прискорить обмін інформації, але основним розвитком є перетворення на платформу для запуску додатків, що будуть працювати швидше за допомогою даного методу, а також випуск повноцінної бібліотеки для розробників[15].

5.4.3. Конкурентні переваги рішення.

Серед основних критеріїв у конкурентоздатності є:

- Патентоздатність (новизна проекту);
- Раціональність виробничої організаційної структури схеми;
- Конкуренто спроможність персоналу;

- Прогресивність технології;
- Прогресивність технологічних процесів та обладнання;
- Науковий рівень розвитку розробників;
- Науковий рівень системи.

Технологічні переваги методу(проекту):

Програмне забезпечення - це програмний модуль, що працює з даними та програмує послідовність виконання операцій програмою у відповідних потоках. В процесі виконання коду програми, модуль зіставлення прогнозів динамічно надає програмі дані для обробки та обраховує їх. Суть методу у реалізації динамічного збору та оброки інформації.

Реалізація у вигляді динамічної бібліотеки з тестовим пакетом, що надасть змогу до імплементації методу на виробництво перевірити результати роботи методу на реальних даних.

Зручний пакет розробника, що надасть додаткову мотивацію до використання методу «вільним» та побічним розробникам.

Програмна реалізація методу має збільшену відмовостійкість, при обриванні з мережею інтернет відключаються лише побічні функції ПЗ, а сам метод розпаралелювання продовжує працювати з тими даними які є на даний момент.

Загальні переваги проекту:

Так як проект та в майбутньому команда(фірма, компанія, тощо) не використовує підхід кредитування тому може дозволити зручний та сталий формат формування цінової політики;

Весь персонал, що буде займатися обслуговуванням та оновленням систем та обладнання у замовників, проходять постійні курси підготовки та розвитку, приймають участь у конференціях;

Оптимізація потужностей, через те що метод прискоренням систем постійно розвивається та знаходиться у постійному процесі збору статистичних

даних, це дає можливість оновлювати ПО та обладнання у найкоротші строки;

Швидка динаміка у оновленнях та підтримці ПО, відповідно до нових статистичних даних зібраних включаючи інші заклади, відповідно чим більше замовників тим більша швидкість оновлення та доповнення методу.

- Відсутність некваліфікованих співробітників;
- Встановлення та оновлення системи прискорення відбувається напряму без посередників;
- Онлайн тестування методу на прикладах[16].

Також слід приділити особливу увагу пункту про тестування, адже при впровадженні подібних методів на виробництво, замовник хоче бути впевненим у тому, що дана система буде давати необхідні результати, це проблема вирішується методом реалізації окремої повноцінної, самостійної користувацької бібліотеки тестування, яка надасть змогу тестування роботи системи на підготовлених даних, побудувати систему покращення роботи підприємства, а саме програмного забезпечення на яке вона встановлюється. Даний аспект надає суттєву перевагу, адже перед встановленням метод може бути повність протестований, а результати покращення проаналізовані замовником[17].

5.4.4 Клієнти та сегменти ринку споживачів

Як вже було зазначено в аналізі зацікавлених сторін, основними клієнтами та споживачами даного методу є підприємства та заклади які займаються криптовалютою. Детальна збірка інформації про клієнтів зібрана у таблиці 4.

Таблиця 4 – Таблиця потенційних клієнтів.

Тип закладу	Опис проблеми закладу, яку вирішує метод	Рейтинг споживача
Компанії, які займаються трейдингом	Подібні заклади оброблюють велику кількість інформацію, яку при цьому треба зберігати та оброблювати, інформації з кожним днем стає все більше і більше, а методи обробки, збереження та надсилання розвиваються занадто повільно, що веде до втрати швидкості та продуктивності роботи закладу та втрати точності зіставлених прогнозів. Спосіб та програмне забезпечення для аналізу та прогнозування криптовалют, частково вирішує дану проблему, адже це і є його суть, прискорити процеси роботи трейдингу та мінімізувати ризики.	10
Трейдери	Зазвичай фізичні особи що маніпулюють валютою для себе.	8

Як видно з таблиці, що наведена вище, основними клієнтами є компанії та фізичні особи, які працюють з криптовалютою. Перевагою даного сегменту є обмін інвестицій та приватність до кожного великого

клієнта, проте мінусом є великі ризики та складність у знаходженні нових клієнтів.

5.5 Унікальна цінність пропозиції (наукового продукту)

Як вже згадувалося раніше, подібних рішень для криптовалюти не існує, тому і було вирішено розробити даний проект.

5.6 Доходи і витрати

5.6.1 Витрати

Для прийняття рішення, щодо інвестиційного проекту, всі витрати, що пов'язані з його здійсненням, необхідно розподілити на інвестиційні та виробничі.

Загальна сума витрат на здійснення проекту включає:

- Витрати на формування основного капіталу містять початкові і поточні інвестиції;
- Витрати на формування оборотного капіталу;
- Виробничі витрати.

Всі інвестиційні потреби підприємства можна поділити на три групи:

- Прямі інвестиції - безпосередньо необхідні для реалізації інвестиційного проекту;
- Супутні інвестиції - вкладення в об'єкти, безпосередньо технології які пов'язані із забезпеченням нормальної експлуатації;
- Інвестування виконання науково-дослідних робіт.

До складу початкових інвестицій відносяться[18]:

- Витрати на дослідження, проведення дослідницьких і конструкторських робіт, на розробку проектних матеріалів, на робоче проектування і прив'язку проекту;
- Витрати на придбання та оренду необхідних сервісів, включаючи вартість підготовки до освоєння;
- Витрати на придбання та доставку машин, обладнання, інструменту та інвентарю, в тому числі імпортних;
- Витрати на приймально-здавальні випробування;
- Витрати на пусконалагоджувальні роботи, комплексне освоєння проектних потужностей і досягнення проектних техніко-економічних показників;
- Витрати на придбання патентів, ліцензій, ноу-хау, технологій та інших амортизаційних нематеріальних активів;
- Витрати на підготовку кадрів для об'єктів, що вводяться в дію;
- Одноразові виплати, зокрема гарантує і страховим організаціям;
- Витрати, що виникають при створенні та реєстрації фірми (оплата юридичних послуг зі складання установчих документів, витрати на реєстрацію фірми і оформлення прав власності;
- Витрати на підготовчі дослідження не враховуючи вартості об'єкта;
- Витрати, пов'язані з діяльністю персоналу в період підготовки виробництва (оплата праці, витрати, утримання приміщень, автомобілів, комп'ютерів та іншого обладнання), не враховані в кошторисної вартості об'єкт.

Висновки по витратам:

Вартість проекту = ціна проектування + ціна розробки + активна підтримка і доопрацювання після впровадження + плата за компоненти та сервіси + реклама і просування. На рисунку 5.6.1 схематично зображено 2 етапи витрат.

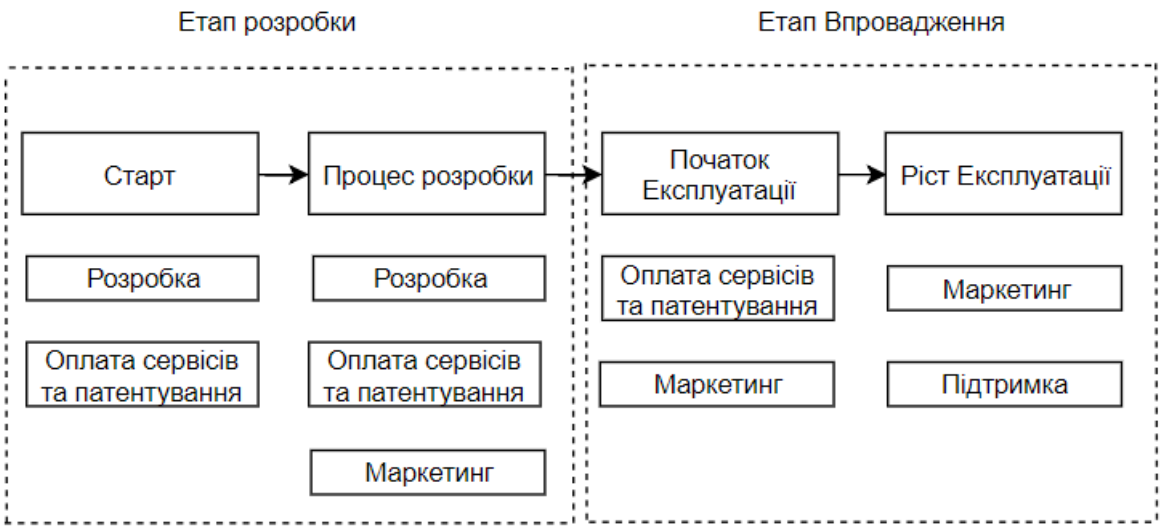

Рис. 28 Види витрат на проект на різних етапах.

5.6.2 Доходи

Як зрозуміло з попередніх пунктів та ринку споживачів, товаром є метод для підвищення ефективності роботи трейдерів та компаній, що працюють з криптовалютою. Основним джерелом доходу будуть кошти з експлуатації даного методу. Загальну дохідність проекту в майбутньому буде прийнято рахувати метриками доходності, такими, як:

Таблиця 5.6.1 описує стадії розвитку проекту, розділи витрат та сумарний дохід. Слід врахувати, що перший та другий етапи є разовими і є препроектими, третій етап розраховується помісячно, після закінчення перших двух етапів.

Таблиця 4. Витрати та доходи проекту

Етап проекту	Статті витрат	Шляхи доходу	Витрати	Дохід	Сумм
Етап розробки	Розробка, оплата сервісів, патентування, маркетинг		2.000 у.о	0	-2.000 у.о
Етап впровадження	Маркетинг, підтримка, реалізація, впровадження	Впровадження у тестовому режимі на виробництво	4.000 у.о	2.000 у.о	-4.000 у.о
Етап експлуатації	Підтримка, обслуговування, розвиток проекту, оплата персоналу	Замовлення, оплата послуг замовниками, підтримка, підписка, оплата тестових режимів, оплата послуг моніторингу закладів, встановлення система обліку інформації	4.000 у.о	10.000 у.о	+6.000 у.о

ВИСНОВКИ

У першому розділі проведений аналіз існуючих підходів аналізу та зіставлення прогнозів валютних ринків. Було проаналізовано їх переваги та недоліки.

Було розглянуто методи аналізу та зіставлення прогнозів та виділено основні різновиди аналізу: фундаментальний та технічний, але надалі буде використано лише підходи до технічного аналізу, так як саме цю частину можна інтерпретувати та автоматизувати за допомогою комп'ютерних алгоритмів. Такі алгоритми дозволять розробити спосіб до аналізу та зіставлення прогнозу на певний проміжок часу.

У другому розділі запропоновано інструменти для прогнозування руху цін за допомогою процесу згортки, в результаті чого ми можемо визнати поведінкові схеми в діаграмі руху цін у реальному часі. Також, знайшовши шаблони, можна порівняти ситуацію на ринку з тим, що трапилося раніше, що дозволяє зробити припущення щодо руху цін у майбутньому.

У третьому розділі описано розроблений програмний комплекс. Докладно описано кожен модуль який використовується в системі, та за яку саме діяльність він відповідає. Наведено діаграми архітектури проекту та скріншоти візуальної частини програмної реалізації. Продемонстровано можливості конфігурації додатків. Архітектура базується на принципах CQRS шаблону проектування, та базується на мікросервісній архітектурі Microsoft Azure. Для роботи всієї системи, повинні бути налаштовані всі окремі сервіси, що є як і плюсом такого підходу, так і недоліком. Оскільки, для налаштування всього проекту з нуля, потрібно провести багато часу за конфігурацію окремо діючих модулів. Які в свою чергу відповідають за різну функціональність проекту.

Розроблений програмний комплекс надає зручний спосіб для моніторингу та перегляду зіставленого прогнозу на певний період часу.

Завдяки розробленому веб додатку, який відображає дані, що були отримані з біржі, та в наслідку оброблені розробленим алгоритмом.

У четвертому розділі було порівняно різні аспекти системи з існуючими методами, а саме обраної нейромережі. Проведено порівняння швидкодії та відмовостійкості. В результаті яких виявилося, що розроблений спосіб перевершує результати нейронної мережі на проміжках часу, в яких відбуваються значні коливання та зміни трендів напрямку ринку. Так як розроблений спосіб працює з фігурами зміну тренду, які краще підходять саме для таких випадків. Щоб досягти подібних результатів за допомогою нейромережі, потрібно провести багато часу за її навчанням та підбором ваг нейронів. Що є досить складною та комплексною роботою, в альтернативу від розробленого способу, який працює завжди з заздалегідь підготовленими шаблонами, та в результаті, дає позитивний результат.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Криптовалюта [Електронний ресурс]. — Режим доступу : https://blockgeeks.com/guides/what-is-cryptocurrency. — Дата доступу : Грудень 2018. — Назва з екрана.
2. Основи технічного аналізу [Електронний ресурс]. — Режим доступу : https://pidruchniki.com/72559/finansi/osnovi_tehnichnogo_analizu . — Дата доступу : Грудень 2018. — Назва з екрана.
3. Сельцовський В.Л. Економіко-статистичні методи аналізу зовнішньої торгівлі - М . Фінанси і статистика, 2004. - 512 с.
4. The Addison-Wesley series in economics, 2004.
5. Niederhoffer V., Kenner L. Practical Speculation. Wiley, 2007.
6. Mak D.K. Science of Financial Market Trading. World Scientific, 2003.
7. Mak D.K. Mathematical techniques in financial market trading. World Scientific, 2006.
8. Технічний аналіз фондового ринку [Електронний ресурс]. — Режим доступу : https://stud.com.ua/50044/finansi/tehnichniy_analiz_fondovogo_rinku. — Дата доступу : Грудень 2018. — Назва з екрана.
9. Быстрое преобразование Фурье [Електронний ресурс]. — Режим доступу : http://ru.dsplib.org/content/fft_introduction/fft_introduction.html. — Дата доступу : Грудень 2018. — Назва з екрана.
10. Azure Best Practices [Електронний ресурс]. — Режим доступу : https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-best-practices. — Дата доступу : Грудень 2018. — Назва з екрана.
11. Windows Virtual Machines Documentation [Електронний ресурс]. — Режим доступу : https://docs.microsoft.com/en-us/azure/virtual-machines/windows/. — Дата доступу : Грудень 2018. — Назва з екрана.

12. Кононюк А. Ю. *Нейронні мережі* та *генетичні алгоритми*. Кононюк. - К. *Корнійчук, 2008. – 446* с.
13. Хайкін С. Нейронні мережі: Повний курс. Пер. с англ. Н. Н. Куссуль, А. Ю. Шелестова. изд., испр. — М.: Издательский дом Вильямс, 2008 - 1103 с.
14. Використання нейронних мереж [Електронний ресурс]. — Режим доступу : https://ukrbukva.net/page,2,92419-Ispol-zovanie-neiyronnyh-seteiy-v-sisteme-Matlab.html
15. Нейронні мережі [Електронний ресурс]. — Режим доступу : https://futurum.today/shtuchni-neironni-merezhi-shcho-tse-take/
16. Microsoft Azure [Електронний ресурс]. — Режим доступу : https://azure.microsoft.com/en-us. — Дата доступу : Грудень 2018. — Назва з екрана.
17. Business [Електронний ресурс]. — Режим доступу : https://en.wikipedia.org/wiki/Business — Дата доступу : Грудень 2018. — Назва з екрана.
18. Поняття і структура доходів підприємства [Електронний ресурс]. — Режим доступу : https://stud.com.ua/49290/ekonomika/ponyattya_struktura_dohodiv_pidpriyemstva — Дата доступу : Грудень 2018. — Назва з екрана.:

ДОДАТКИ

ЛІСТИНГ КОДУ

```
namespace CryptocurrencyForecasting.WebJob
{
    class Program
    {
        public static async Task Main(string[] args)
        {
            var builder = new HostBuilder()
                .UseEnvironment("Dev")
                .ConfigureWebJobs(b =>
                {
                    b.AddAzureStorageCoreServices()
                    .AddTimers();
                })
                .ConfigureAppConfiguration(b =>
                {
                    b.AddJsonFile("appsettings.json", optional: false, reloadOnChange:
true);
                    b.AddCommandLine(args);
                })
                .ConfigureLogging((context, b) =>
                {
                    b.SetMinimumLevel(LogLevel.Debug);
                    b.AddConsole();

                })
                .ConfigureServices((context, services) =>
                 {
                    services.AddSingleton<INameResolver, NameResolver>(_ => new
NameResolver(context.Configuration, new[] { "Values" }));
                     services.AddTransient<Functions, Functions>();
                 })
                .UseConsoleLifetime();

            var host = builder.Build();
            using (host)
            {
                await host.RunAsync();
            }
        }
    }
}

namespace CryptocurrencyForecasting.WebJob.Models
{
    public class CalcResult : TableEntity
    {

        public Status Status { get; set; }
        public DateTime Date { get; set; }
    }


    public enum Status
    {
        Unknown,
        Up,
        Down,
        Stay
    }
}

namespace CryptocurrencyForecasting.WebJob.Models
{
```

```
    public class Candle
    {
        public double Max { get; set; }
        public double Min { get; set; }
        public DateTime Date { get; set; }
    }
}

namespace CryptocurrencyForecasting.WebJob.Models
{
    public class Feed : TableEntity
    {
        public double Ask { get; set; }
        public double Bid { get; set; }
        public double Last { get; set; }
        public DateTime Date { get; set; }
    }
}

namespace CryptocurrencyForecasting.WebJob.Helpers
{
    public class NameResolver : INameResolver
    {
        private readonly IConfiguration _configuration;
        private readonly string[] _paths;

        public NameResolver(IConfiguration configuration, string[] paths)
        {
            _configuration = configuration;
            _paths = paths == null ? new string[] { } : paths.Where(_ =>
!string.IsNullOrWhiteSpace(_)).ToArray();
        }

        public string Resolve(string name)
        {
            return _paths.Select(path =>
_configuration.GetSection($"{path}:{name}").Value).FirstOrDefault(val =>
!string.IsNullOrWhiteSpace(val));
        }
    }
}

using CryptocurrencyForecasting.Contracts;
using CryptocurrencyForecasting.WebJob.Models;
using Microsoft.Azure.WebJobs;
using Microsoft.Extensions.Logging;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Auth;
using Microsoft.WindowsAzure.Storage.Table;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Threading.Tasks;

namespace CryptocurrencyForecasting.WebJob
{
    public partial class Functions
    {

        [Singleton]
        public async Task GetMarketData([TimerTrigger("%Timer%", RunOnStartup = true)]
TimerInfo timerInfo, ILogger logger)
```

```
        {


            System.Threading.Thread.Sleep(1000);
            logger.LogInformation("Trying to get data from exchange...");


            HttpClient client = new HttpClient();
            client.BaseAddress = new Uri("https://bittrex.com");
            client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));
            HttpResponseMessage response =
client.GetAsync("https://bittrex.com/api/v1.1/public/getticker?market=usd-
btc").Result;

            logger.LogInformation("Prepare request for https://bittrex.com...");

            BitTrex feedItem = null;
            if (response.IsSuccessStatusCode)
            {
                try
                {
                    logger.LogInformation("Parsing data from exchange...");

                    var result = response.Content.ReadAsStringAsync().Result;
                    feedItem = JsonConvert.DeserializeObject<BitTrex>(result);

                    logger.LogInformation("Data successfully parsed...");
                }
                catch (Exception ex)
                {
                    logger.LogInformation($"Error during parsing : {ex.Message}");
                }



                CloudStorageAccount storageAccount = new CloudStorageAccount(new
StorageCredentials("cryptforecast",
"c5CD/ep7+riweoGQit7DKucCJw2Gy42HxcGA++aS00CxMp1GZXxGFpyenORXztC6LpUNZgxc/ltcgdNGRVrQE
A=="), true);

                var uri = storageAccount.TableStorageUri.PrimaryUri;
                var cred = storageAccount.Credentials;
                // Create the table client.

                CloudTableClient tableClientWithSAS =
storageAccount.CreateCloudTableClient();
                TableQuery<Feed> query = new TableQuery<Feed>();

                CloudTable peopleTable =
tableClientWithSAS.GetTableReference("feedTest");

                var items = await peopleTable.ExecuteQuerySegmentedAsync(query, null);



                var model = feedItem.Result;

                try
                {
                    logger.LogInformation("Try to save data from exchange...");

                    Feed customer1 = new Feed()
                    {
```

```
                        PartitionKey = string.Format("{0}-{1}-'{2}'", model.Bid,
model.Ask, DateTime.Now.ToString()),
                        RowKey = "USD-BTC",
                        Ask = model.Ask,
                        Bid = model.Bid,
                        Last = model.Last,
                        Date = DateTime.Now
                    };




                    // Create the TableOperation object that inserts the customer
entity.
                    TableOperation aaa = TableOperation.Insert(customer1);

                    // Execute the insert operation.
                    await peopleTable.ExecuteAsync(aaa);




                    logger.LogInformation("Data successfully saved...");
                }
                catch (Exception ex)
                {
                    logger.LogInformation($"Error during saving : {ex.Message}");
                }

            }

        }

        [Singleton]
        public async Task CalcData([TimerTrigger("%Timer%", RunOnStartup = true)]
TimerInfo timerInfo, ILogger logger)
        {

            const int minStep = 3;

            CloudStorageAccount storageAccount = new CloudStorageAccount(new
StorageCredentials("cryptforecast",
"c5CD/ep7+riweoGQit7DKucCJw2Gy42HxcGA++aS00CxMp1GZXxGFpyenORXztC6LpUNZgxc/ltcgdNGRVrQE
A=="), true);

            var uri = storageAccount.TableStorageUri.PrimaryUri;
            var cred = storageAccount.Credentials;
            // Create the table client.

            CloudTableClient tableClientWithSAS =
storageAccount.CreateCloudTableClient();
            TableQuery<Feed> query = new TableQuery<Feed>();

            CloudTable table = tableClientWithSAS.GetTableReference("feedTest");

            var items = await table.ExecuteQuerySegmentedAsync(query, null);
            var groupByDate = items.OrderBy(i => i.Date).GroupBy(i => i.Date.Date);

            List<Feed> list = new List<Feed>();

            List<Candle> candleList = new List<Candle>();

            foreach (var item in groupByDate)
```

```
            {
                var oredered = item.OrderBy(i => i.Bid);

                var min = item.First();
                var max = item.Last();

                candleList.Add(new Candle()
                {
                    Date = min.Date.Date,
                    Max = max.Bid,
                    Min = min.Bid
                });
            }

            var calcRes = new List<CalcResult>();

            const int threshold = 100;



            int step = 0;

            List<Candle> collList = new List<Candle>();
            ///algoritm
            ///
            foreach (var candle in candleList)
            {

                collList.Add(candle);

                if (collList.Count == minStep - 1)
                {
                    var startMax = collList.First().Max;
                    var endMax = collList.Last().Max;

                    var startMin = collList.First().Min;
                    var endMin = collList.Last().Min;

                    //check if in range

                    foreach (var item in collList)
                    {
                        if (collList.IndexOf(item) == 0) continue;

                        if (item.Max > startMax + threshold || item.Max > endMax +
threshold)
                        {
                            break;
                        }

                        if (item.Min < startMin - threshold || item.Min < endMin -
threshold)
                        {
                            break;
                        }

                        if (collList.IndexOf(item) ==
collList.IndexOf(collList.Last()))
                        {
                            Status status = Status.Unknown;

                            if (startMax > endMax && startMin > endMin || startMin >
endMin || startMax > endMax)
                            {
                                status = Status.Up;
```

```
                            }

                            if (startMax < endMax && startMin < endMin || startMax >
endMax)
                            {
                                status = Status.Down;
                            }


                            if ((startMax - endMax) * -1 < threshold && (startMin -
endMin) * -1 < threshold)
                            {
                                status = Status.Stay;
                            }

                            calcRes.Add(
                                    new CalcResult()
                                    {
                                        Date = item.Date.AddDays(2),
                                        Status = status
                                    }
                            );
                        }
                    }


                    collList.Remove(collList.First());
                    step--;
                }

                step++;
            }


            CloudTable calcFeedTable =
tableClientWithSAS.GetTableReference("calcfeed");



            foreach (var calcItem in calcRes)
            {
                try
                {
                    calcItem.PartitionKey = calcItem.Status.ToString();
                    calcItem.RowKey = calcItem.Date.ToString();
                    TableOperation insertOperation = TableOperation.Insert(calcItem);
                    await calcFeedTable.ExecuteAsync(insertOperation);
                    logger.LogInformation("Forecast saved...");
                }
                catch
                {
                    logger.LogInformation($"Forecast already exists...");
                }

            }


        }

    }
}
```